



PIC18F45J10 系列
数据手册

采用纳瓦技术的 **28/40/44** 引脚
高性能 **RISC** 单片机

请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展之中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

提供本文档的中文版本仅为了便于理解。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。未经 Microchip 书面批准, 不得将 Microchip 的产品用作生命维持系统中的关键组件。在 Microchip 知识产权保护下, 不得暗或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Accuron、dsPIC、KEELOQ、microID、MPLAB、PIC、PICmicro、PICSTART、PRO MATE、PowerSmart、rfPIC 和 SmartShunt 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

AmpLab、FilterLab、Migratable Memory、MXDEV、MXLAB、PICMASTER、SEEVAL、SmartSensor 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、dsPICDEM、dsPICDEM.net、dsPICworks、ECAN、ECONOMONITOR、FanSense、FlexROM、fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、Linear Active Thermistor、MPASM、MPLIB、MPLINK、MPSIM、PICKit、PICDEM、PICDEM.net、PICLAB、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、Real ICE、rfLAB、rfPICDEM、Select Mode、Smart Serial、SmartTel、Total Endurance、UNI/O、WiperLock 和 Zena 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2005, Microchip Technology Inc. 版权所有。

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 及位于加利福尼亚州 Mountain View 的全球总部、设计中心和晶圆生产厂均于 2003 年 10 月通过了 ISO/TS-16949:2002 质量体系认证。公司在 PICmicro® 8 位单片机、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

采用纳瓦技术的 28/40/44 引脚 高性能 RISC 单片机

单片机的特殊性能:

- 工作电压范围: 2.0V 至 3.6V
- 可接受 5.5V 输入 (仅数字引脚)
- 片上 2.5V 稳压器
- 低功耗的高速 CMOS 闪存技术
- 优化的 C 编译器架构:
 - 为优化重入代码而设计的可选的扩展指令集
- 中断优先级
- 8 x 8 单周期硬件乘法器
- 扩展的看门狗定时器 (Watchdog Timer, WDT):
 - 可编程周期从 4 ms 到 131s
- 通过两个引脚进行单电源供电的在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 通过两个引脚采用三个断点进行在线调试 (In-Circuit Debug, ICD)
- 功耗管理模式:
 - 运行: CPU 打开, 外设打开
 - 空闲: CPU 关闭, 外设打开
 - 休眠: CPU 关闭, 外设关闭

灵活的振荡器结构:

- 两种晶振模式, 频率最高为 40 MHz
- 两种外部时钟模式, 频率最高为 40MHz
- 内部 31 kHz 振荡器
- 辅助振荡器使用 Timer1 (工作频率为 32kHz)
- 双速振荡器起振
- 故障保护时钟监视器:
 - 当外设时钟停止时可使器件安全关闭

外设特点:

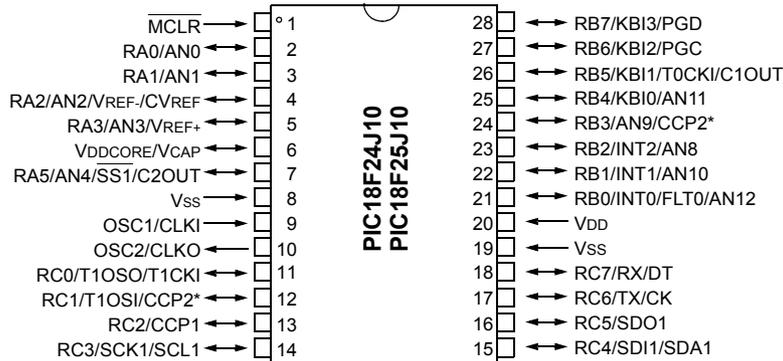
- 高灌 / 拉电流 25 mA/25 mA (PORTB 和 PORTC)
- 3 个可编程外部中断
- 4 个输入电平变化中断
- 一个捕捉 / 比较 / PWM (CCP) 模块
- 一个增强型捕捉 / 比较 / PWM (ECCP) 模块:
 - 1 路、2 路或 4 路 PWM 输出
 - 可选择的极性
 - 可编程的死区时间
 - 自动关闭和自动重启
- 2 个主控同步串行端口 (Master Synchronous Serial Port, MSSP) 模块, 支持 3 线 SPI (共 4 种模式) 和 I²C™ 主 / 从模式
- 一个增强型可寻址 USART 模块:
 - 支持 RS-485、RS-232 和 LIN 1.2
 - 起始位自动唤醒
 - 自动波特率检测
- 10 位最多 13 路通道的数 / 模转换器模块 (A/D):
 - 自动采集功能
 - 可在休眠模式下进行转换
 - 自动校准功能
- 输入复用的双模拟比较器

器件	程序存储器		SRAM 数据存储器 (字节)	I/O	10 位 A/D (通道数)	CCP/ ECCP (PWM)	MSSP		EUSART	比较器	8/16 位定时器	
	闪存 (字节)	单字指令数					SPI	主控 I ² C™				
PIC18F24J10	16K	8192	1024	21	10	2/0	1	有	有	1	2	1/2
PIC18F25J10	32K	16384	1024	21	10	2/0	1	有	有	1	2	1/2
PIC18F44J10	16K	8192	1024	32	13	1/1	2	有	有	1	2	1/2
PIC18F45J10	32K	16384	1024	32	13	1/1	2	有	有	1	2	1/2

PIC18F45J10 系列

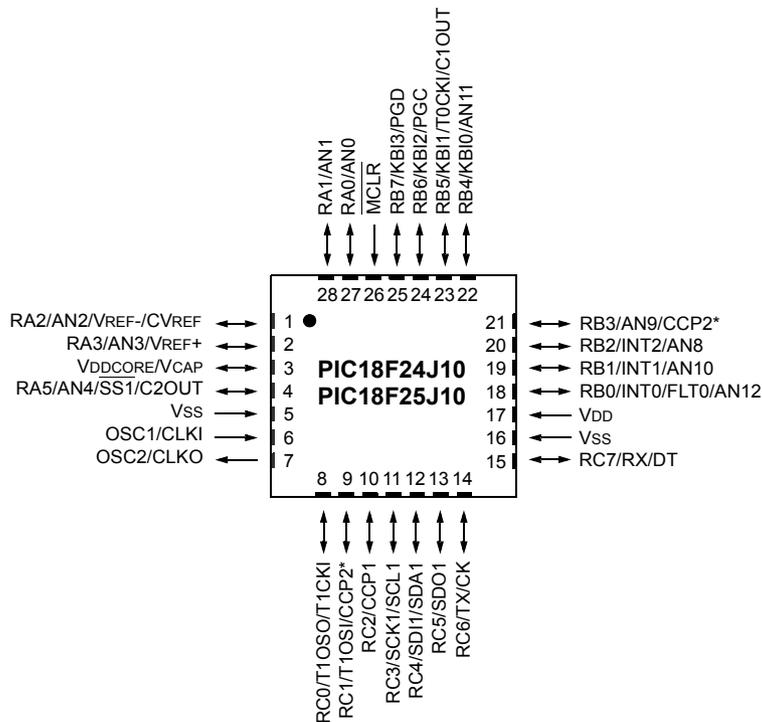
引脚图

28 引脚 SPDIP、SOIC 和 SSOP (300 MIL)



* 引脚功能取决于器件配置。

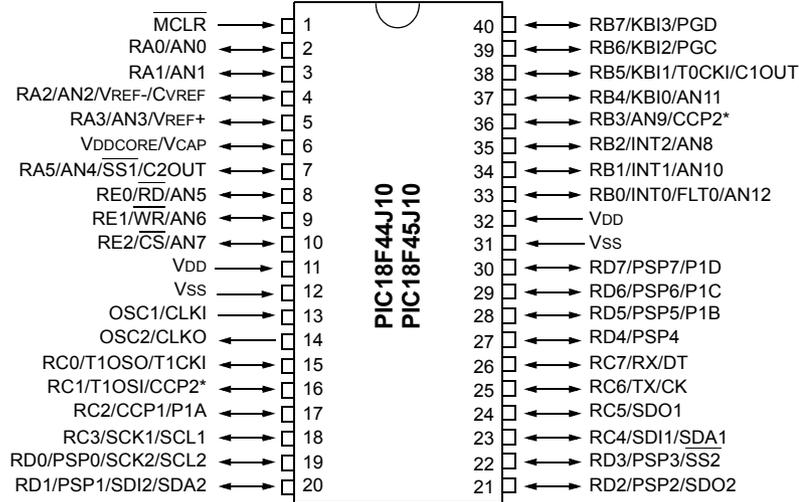
28 引脚 QFN



* 引脚功能取决于器件配置。

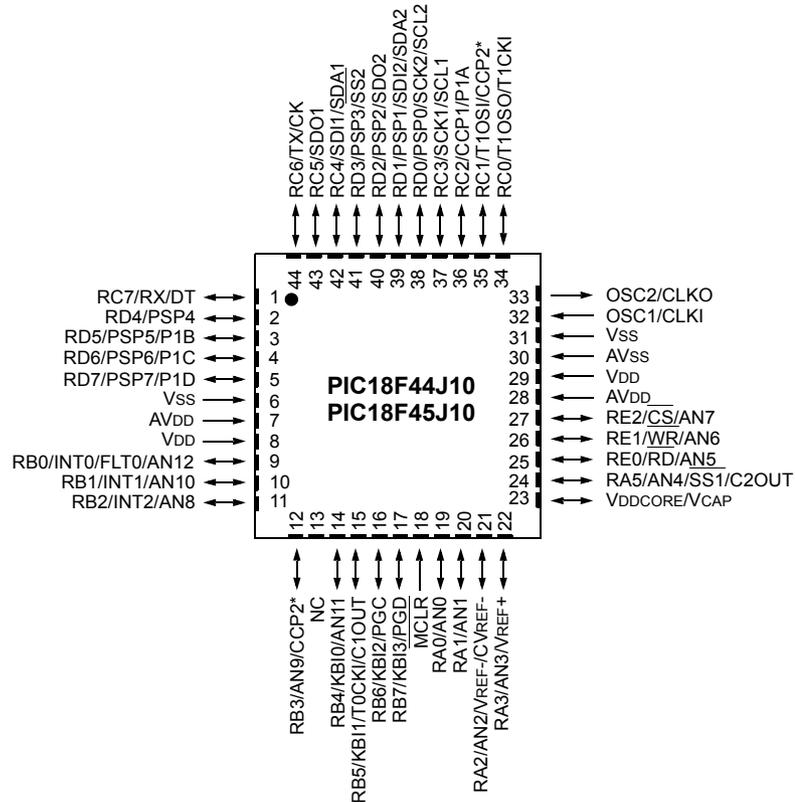
引脚图 (续)

40 引脚 PDIP (600 MIL)



* 引脚功能取决于器件配置。

44 引脚 QFN

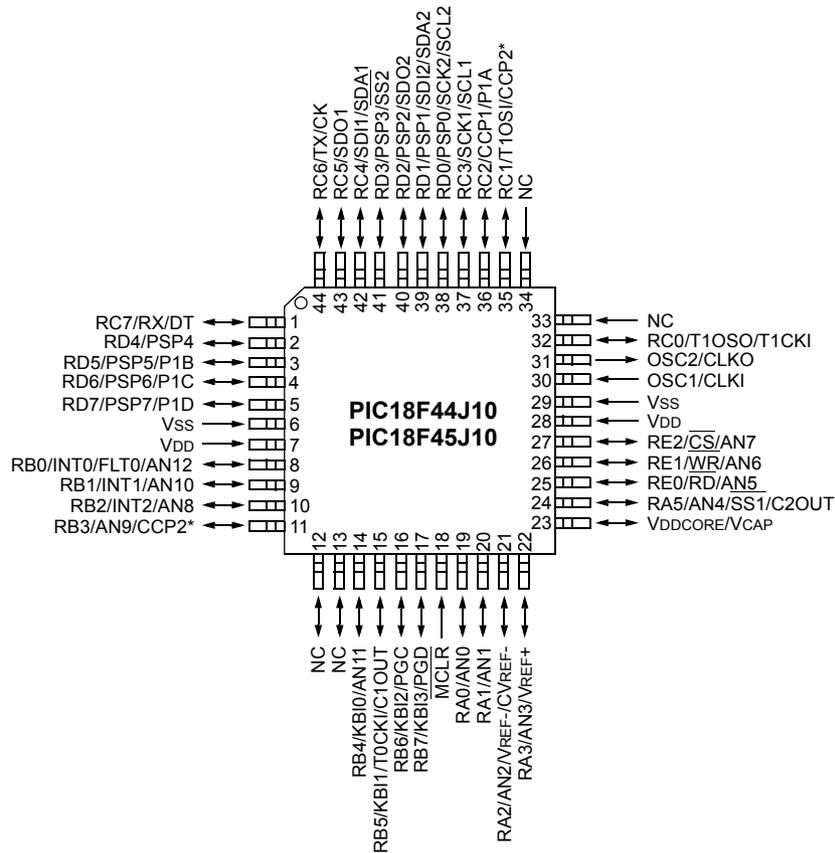


* 引脚功能取决于器件配置。

PIC18F45J10 系列

引脚表 (续)

44 引脚 TQFP



* 引脚功能取决于器件配置。

目录

1.0	器件概述	7
2.0	振荡器配置	23
3.0	功耗管理模式	31
4.0	复位	37
5.0	存储器构成	47
6.0	闪存程序存储器	67
7.0	8 x 8 硬件乘法器	77
8.0	中断	79
9.0	I/O 端口	93
10.0	Timer0 模块	111
11.0	Timer1 模块	115
12.0	Timer2 模块	121
13.0	捕捉 / 比较 / PWM (CCP) 模块	123
14.0	增强型捕捉 / 比较 / PWM (ECCP) 模块	131
15.0	主控同步串口 (MSSP) 模块	145
16.0	增强型通用同步 / 异步收发器 (EUSART)	187
17.0	10 位模数转换器 (A/D) 模块	209
18.0	比较器模块	219
19.0	比较器参考电压源模块	225
20.0	CPU 的特殊性能	229
21.0	指令集综述	241
22.0	开发支持	291
23.0	电气规范	297
24.0	DC 和 AC 特性图表	331
25.0	封装信息	333
附录 A:	版本历史	343
附录 B:	高档器件系列间的移植	343
索引		345
Microchip 网站		355
客户变更通知服务		355
客户支持		355
读者反馈		356
PIC18F45J10 系列产品标识体系		357

PIC18F45J10 系列

致客户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的要求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 CTRC@microchip.com，或将本数据手册后附的《读者反馈表》传真到 86-21-5407 5066。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中数字串后的字母是版本号，例如 DS30000A 是 DS30000 的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站：<http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 www.microchip.com 上注册。

1.0 器件概述

该文档包含以下器件特定的信息:

- PIC18F24J10
- PIC18F25J10
- PIC18F44J10
- PIC18F45J10
- PIC18LF24J10
- PIC18LF25J10
- PIC18LF44J10
- PIC18LF45J10

PIC18F45J10 系列除了具有 PIC18 单片机的传统优点,即出色的计算性能和经济实惠的价格之外,还引进了一些增强的功能,使其成为许多高性能、低功耗应用的理想选择。

1.1 新的内核特性

1.1.1 纳瓦技术

PIC18F45J10 系列的所有器件都具有一系列能显著降低工作功耗的特性。主要包括以下几项:

- **备用运行模式:** 通过将 Timer1 或内部振荡器模块作为单片机时钟源,可使代码执行时的功耗大约降低 90%。
- **多种空闲模式:** 单片机还可在其 CPU 内核禁止而外设工作的情况下运行。处于这些状态时,功耗能降得更低,只有正常工作需求的 4%。
- **动态模式切换:** 在器件工作期间可由用户代码调用该功耗管理模式,允许用户将节能的理念融入他们的应用软件设计中。
- **关键模块低功耗:** Timer1 和看门狗定时器的功耗需求最低。请参见第 23.0 节“电气规范”了解具体数值。

1.1.2 振荡器选项和特性

PIC18F45J10 系列的所有器件提供 3 种不同的振荡器选项,包括:

- 一个晶振模式,使用晶振或陶瓷谐振器
- 一个外部时钟模式
- INTRC 时钟源 (约 31 kHz)

除了可用作时钟源外,内部振荡器模块还提供了一个稳定的参考源,用以实现下列额外的功能以使器件更安全可靠地工作:

- **故障保护时钟监视器:** 该部件不停地监视主时钟源,将其与内部振荡器提供的参考信号作比较。如果发生了时钟故障,单片机会切换到内部振荡器模块,使器件可继续低速工作或安全关闭。
- **双速启动:** 该功能允许在上电复位或从休眠模式唤醒时将内部振荡器用作时钟源,直到主时钟源可用为止。

PIC18F45J10 系列

1.2 其他特殊性能

- **通信:** PIC18F45J10 系列包含了一系列串行通信外设, 包括一个独立增强型 USART 和 2 个主控 SSP 模块, 具备 SPI 和 I²C (主控和从动) 两种工作模式。此外, 一个通用 I/O 端口可以被重新配置为 8 位并行从动端口, 实现处理器到处理器的直接通信。
- **自编程性:** 这些器件能在内嵌软件控制下对各自的程序存储空间进行写操作。通过使用引导加载子程序, 可以创建能够实现现场自我更新的应用程序。
- **扩展指令集:** PIC18F45J10 系列在 PIC18 指令集的基础上进行了扩展, 添加了 8 条新指令和变址寻址模式。此扩展可以使用一个器件配置选项使能, 它是为优化重入代码而特别设计的, 这些代码是使用高级语言 (如 C 语言) 开发的。
- **增强型 CCP 模块:** 在 PWM 模式下, 该模块提供 1、2 或 4 路调制输出来控制半桥和全桥驱动器。其他功能包括自动关闭 (自动关闭能在中断或其他条件下禁止 PWM 输出) 和自动重启 (自动重启能在禁止条件被清除时再次激活输出)。
- **增强型可寻址 USART:** 该串行通信模块可进行标准的 RS-232 通信并支持 LIN 总线协议。其他增强功能包括自动波特率检测和精度更高的 16 位波特率发生器。
- **10 位 A/D 转换器:** 该模块实现了可编程采集时间, 从而不必在选择通道和启动转换之间等待一个采样周期, 因而减少了代码开销。
- **扩展的看门狗定时器 (WDT):** 该增强的看门狗定时器添加了 16 位预分频器, 扩展了超时周期范围, 该超时周期在整个工作电压和温度范围内保持稳定。请参见第 23.0 节 “电气规范” 了解超时周期。

1.3 系列中各产品的具体信息

PIC18F45J10 系列器件具有 28 引脚和 40/44 引脚两种封装形式。图 1-1 和图 1-2 分别为这两类器件的框图。这两类器件在以下 5 个方面存在差异:

1. 闪存程序存储器 (PIC18F24J10/44J10 器件为 16KB, PIC18F25J10/45J10 为 32KB)。
2. A/D 通道 (28 引脚器件有 10 个, 40/44 引脚器件有 13 个)。
3. I/O 端口 (28 引脚器件上有 3 个双向端口, 40/44 引脚器件上有 5 个双向端口)。
4. CCP 和增强型 CCP (28 引脚器件有 2 个标准的 CCP 模块, 40/44 引脚器件有 1 个标准的 CCP 模块和 1 个 ECCP 模块)。
5. 并行从动端口 (只存在于 40/44 引脚器件)。
6. PIC18F24J10/25J10 器件具备 1 个 MSSP 模块, PIC18F44J10/45J10 器件具备 2 个 MSSP 模块。
7. 器件编号中带 “F” 的器件 (如 PIC18F25J10) 的 VDD 最小值可为 2.8V, 而器件编号中带 “LF” 的器件 (如 PIC18LF25J10) 能在 2.0-3.6 V 的 VDD 范围内工作, 但是 VDDCORE 决不能超过 VDD。

该系列器件的其他功能都是相同的。表 1-1 汇总了这些功能。

表 1-2 和表 1-3 列举了所有器件的引脚排列方式。

PIC18F45J10 系列器件具有片内稳压器, 可向内核提供稳定的电平。器件编号中带有 “F” 的器件 (如 PIC18F25J10) 使能了稳压器。这些器件能在 2.7-3.6 V 的 VDD 范围内运行, 但是要在 VDDCORE 引脚和 VSS 引脚之间连接低 ESR 的电容。器件编号中带 “LF” 的器件 (如 PIC18LF24J10) 没有使能稳压器。应给 VDDCORE 引脚施加 2.0-2.7 V 的电压, 而给 VDD 引脚施加 2.0-3.6 V 的电压 (VDDCORE 决不能超过 VDD)。请参见第 20.3 节 “片内稳压器” 了解内部稳压器的详细信息。

PIC18F45J10 系列

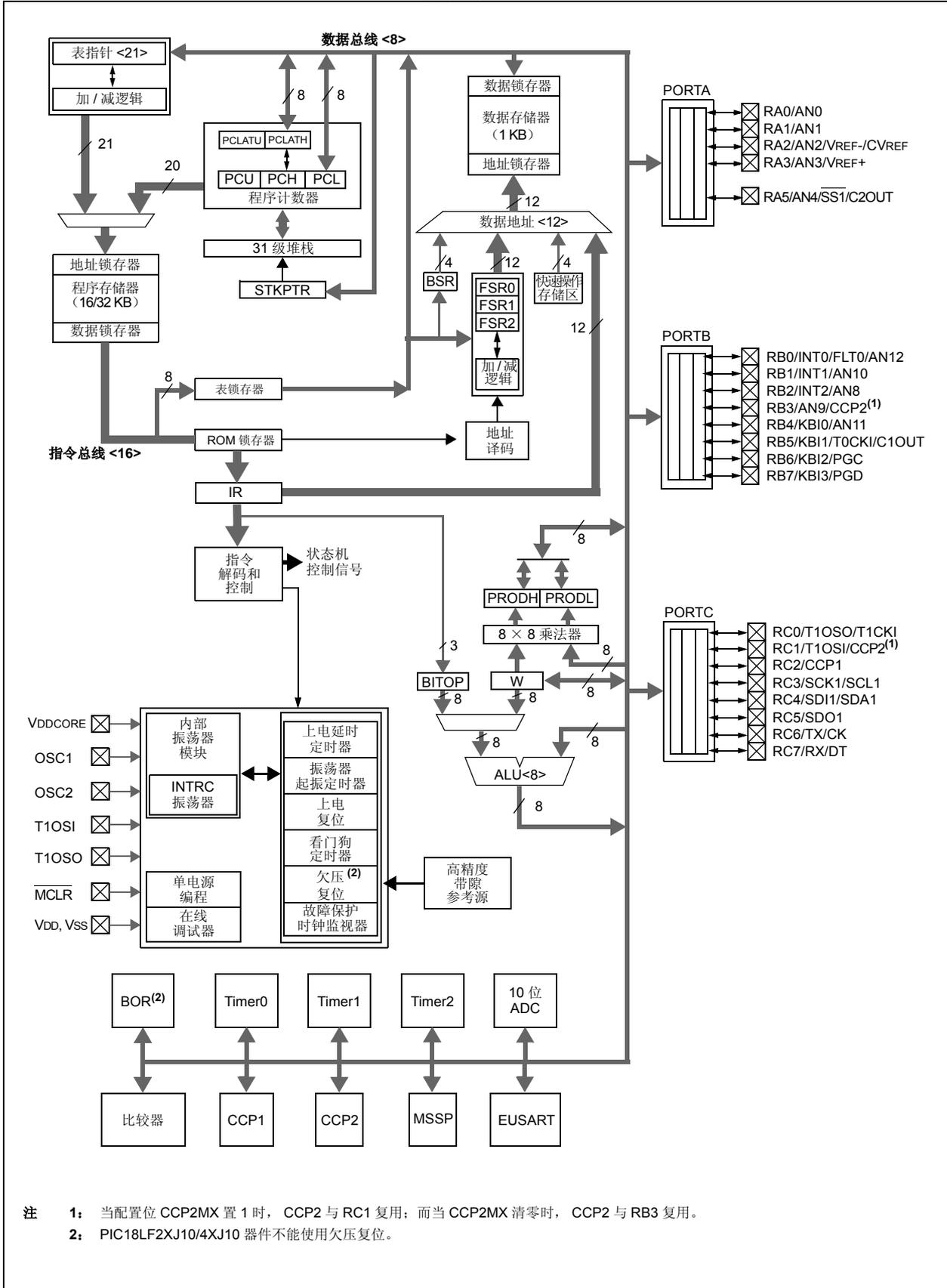
表 1-1: 器件特性

特性	PIC18F24J10	PIC18F25J10	PIC18F44J10	PIC18F45J10
工作频率	DC – 40 MHz			
程序存储器 (字节)	16384	32768	16384	32768
程序存储器 (指令)	8192	16384	8192	16384
数据存储器 (字节)	768	1536	768	1536
中断源	19	19	20	20
I/O 端口	端口 A、B 和 C	端口 A、B 和 C	端口 A、B、C、D 和 E	端口 A、B、C、D 和 E
定时器	3	3	3	3
捕捉 / 比较 / PWM 模块	2	2	1	1
增强型捕捉 / 比较 / PWM 模块	0	0	1	1
串行通信	MSSP, 增强型 USART	MSSP, 增强型 USART	MSSP, 增强型 USART	MSSP, 增强型 USART
并行通信 (PSP)	无	无	有	有
10 位模数转换模块	10 路输入通道	10 路输入通道	13 路输入通道	13 路输入通道
复位 (和延迟)	POR, BOR ⁽¹⁾ , RESET 指令, 堆栈满, 堆栈下溢 (PWRT 和 OST), MCLR 和 WDT	POR, BOR ⁽¹⁾ , RESET 指令, 堆栈满, 堆栈下溢 (PWRT 和 OST), MCLR 和 WDT	POR, BOR ⁽¹⁾ , RESET 指令, 堆栈满, 堆栈下溢 (PWRT 和 OST), MCLR 和 WDT	POR, BOR ⁽¹⁾ , RESET 指令, 堆栈满, 堆栈下溢 (PWRT 和 OST), MCLR 和 WDT
可编程欠压复位	有	有	有	有
指令集	75 条指令; 使能了扩展指令集 后总共为 83 条指令			
封装	28 引脚 SPDIP 28 引脚 SOIC 28 引脚 SSOP 28 引脚 QFN	28 引脚 SPDIP 28 引脚 SOIC 28 引脚 SSOP 28 引脚 QFN	40 引脚 PDIP 44 引脚 QFN 44 引脚 TQFP	40 引脚 PDIP 44 引脚 QFN 44 引脚 TQFP

注 1: PIC18LF2XJ10/4XJ10 器件不能使用欠压复位。

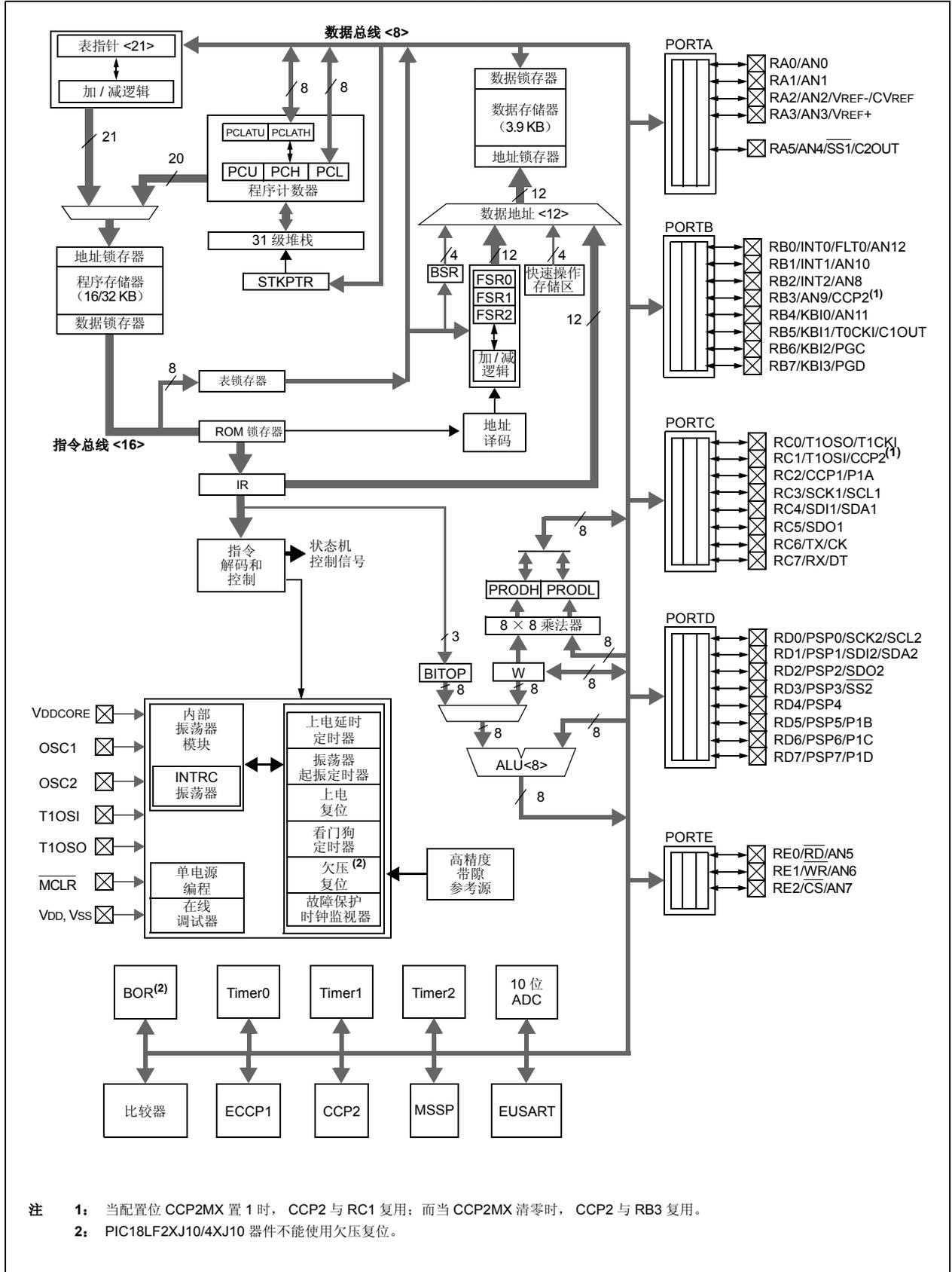
PIC18F45J10 系列

图 1-1: PIC18F24J10/25J10 (28 引脚) 框图



注 1: 当配置位 CCP2MX 置 1 时, CCP2 与 RC1 复用; 而当 CCP2MX 清零时, CCP2 与 RB3 复用。
 2: PIC18LF2XJ10/4XJ10 器件不能使用欠压复位。

图 1-2: PIC18F44J10/45J10 (40/44 引脚) 框图



PIC18F45J10 系列

表 1-2: PIC18F24J10/25J10 I/O 引脚排列说明

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	SPDIP, SOIC, SSOP	QFN			
MCLR MCLR	1	26	I	ST	主清零（输入）或编程电压（输入）。 主清零（复位）输入。此引脚为低电平时，器件复位。
OSC1/CLKI OSC1 CLKI	9	6	I I	— CMOS	振荡器晶振或外部时钟输入。 振荡器晶振输入或外部时钟源输入。 外部时钟源输入。总是与 OSC1 引脚功能复用。请参见相关的 OSC2/CLKO 引脚。
OSC2/CLKO OSC2 CLKO	10	7	O O	— —	振荡器晶振或时钟输出。 振荡器晶振输出。在晶振模式下，该引脚与晶振或谐振器相连。 在 EC 模式下，OSC2 引脚输出 CLKO 振荡信号，该信号是 OSC1 引脚上振荡信号的 4 分频，该频率等于指令周期的倒数。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

注 1: 当配置位 CCP2MX 置 1 时，对 CCP2 进行默认分配。
注 2: 当配置位 CCP2MX 清零时，对 CCP2 进行其他分配。

表 1-2: PIC18F24J10/25J10 I/O 引脚排列说明 (续)

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	SPDIP, SOIC, SSOP	QFN			
RA0/AN0 RA0 AN0	2	27	I/O I	TTL 模拟	PORTA 是双向 I/O 端口。 数字 I/O。 模拟输入 0。
RA1/AN1 RA1 AN1	3	28	I/O I	TTL 模拟	数字 I/O。 模拟输入 1。
RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF	4	1	I/O I I O	TTL 模拟 模拟 模拟	数字 I/O。 模拟输入 2。 A/D 参考电压 (低电平端) 输入。 比较器参考电压输出。
RA3/AN3/VREF+ RA3 AN3 VREF+	5	2	I/O I I	TTL 模拟 模拟	数字 I/O。 模拟输入 3。 A/D 参考电压 (高电平端) 输入。
RA5/AN4/SS1/C2OUT RA5 AN4 SS1 C2OUT	7	4	I/O I I O	TTL 模拟 TTL —	数字 I/O。 模拟输入 4。 SPI 从动选择输入。 比较器 2 输出。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出

CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。
注 2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

PIC18F45J10 系列

表 1-2: PIC18F24J10/25J10 I/O 引脚排列说明 (续)

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	SPDIP, SOIC, SSOP	QFN			
RB0/INT0/FLT0/AN12 RB0 INT0 FLT0 AN12	21	18	I/O 	TTL ST ST 模拟	PORTB 是双向 I/O 端口。PORTB 在所有的输入端都可以软件编程为内部弱上拉。 数字 I/O。 外部中断 0。 CCP1 模块 PWM 故障输入。 模拟输入 12。
RB1/INT1/AN10 RB1 INT1 AN10	22	19	I/O 	TTL ST 模拟	数字 I/O。 外部中断 1。 模拟输入 10。
RB2/INT2/AN8 RB2 INT2 AN8	23	20	I/O 	TTL ST 模拟	数字 I/O。 外部中断 2。 模拟输入 8。
RB3/AN9/CCP2 RB3 AN9 CCP2 ⁽¹⁾	24	21	I/O I/O	TTL 模拟 ST	数字 I/O。 模拟输入 9。 捕捉 2 输入 / 比较器 2 输出 / PWM2 输出。
RB4/KBI0/AN11 RB4 KBI0 AN11	25	22	I/O 	TTL TTL 模拟	数字 I/O。 电平变化中断引脚。 模拟输入 11。
RB5/KBI1/T0CKI/C1OUT RB5 KBI1 T0CKI C1OUT	26	23	I/O O	TTL TTL ST —	数字 I/O。 电平变化中断引脚。 Timer0 外部时钟输入。 比较器 1 输出。
RB6/KBI2/PGC RB6 KBI2 PGC	27	24	I/O I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP™ 编程时钟引脚。
RB7/KBI3/PGD RB7 KBI3 PGD	28	25	I/O I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程数据引脚。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。
注 2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

表 1-2: PIC18F24J10/25J10 I/O 引脚排列说明 (续)

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	SPDIP, SOIC, SSOP	QFN			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST — ST	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1 外部时钟输入。
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 ⁽²⁾	12	9	I/O I I/O	ST 模拟 ST	数字 I/O。 Timer1 振荡器输入。 捕捉 2 输入 / 比较器 2 输出 / PWM2 输出。
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST ST	数字 I/O。 捕捉 1 输入 / 比较器 1 输出 / PWM1 输出。
RC3/SCK1/SCL1 RC3 SCK1 SCL1	14	11	I/O I/O I/O	ST ST ST	数字 I/O。 SPI 模式的同步串行时钟输入 / 输出。 I ² C™ 模式的同步串行时钟输入 / 输出。
RC4/SDI1/SDA1 RC4 SDI1 SDA1	15	12	I/O I I/O	ST ST ST	数字 I/O。 SPI 数据输入。 I ² C 数据 I/O。
RC5/SDO1 RC5 SDO1	16	13	I/O O	ST —	数字 I/O。 SPI 数据输出。
RC6/TX/CK RC6 TX CK	17	14	I/O O I/O	ST — ST	数字 I/O。 EUSART 异步发送。 EUSART 同步时钟 (参见相关 RX/DT 引脚)。
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST ST ST	数字 I/O。 EUSART 异步接收。 EUSART 同步数据 (参见相关 TX/CK 引脚)。
VSS	8, 19	5, 16	P	—	逻辑电路和 I/O 引脚的参考地。
VDD	20	17	P	—	逻辑电路和 I/O 引脚的正电源。
VDDCORE/VCAP VDDCORE VCAP	6	3	P P	— —	逻辑电路和 I/O 引脚的正电源。 逻辑电路和 I/O 引脚的参考地。

图注: TTL = TTL 兼容输入
 ST = CMOS 电平的施密特触发器输入
 O = 输出
 CMOS = CMOS 兼容输入或输出
 I = 输入
 P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。
 2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

PIC18F45J10 系列

表 1-3: PIC18F44J10/45J10 I/O 引脚排列说明

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
MCLR MCLR	1	18	18	I	ST	主清零（输入）或编程电压（输入）。 主清零（复位）输入。此引脚为低电平时，器件复位。
OSC1/CLKI OSC1 CLKI	13	32	30	I I	— CMOS	振荡器晶振或外部时钟输入。 振荡器晶振输入或外部时钟源输入。 外部时钟源输入。总是与 OSC1 引脚功能复用。参见相关的 OSC2/CLKO 引脚。
OSC2/CLKO OSC2 CLKO	14	33	31	O O	— —	振荡器晶振或时钟输出。 振荡器晶振输出。在晶振模式下，该引脚与晶振或谐振器相连。 在 RC 模式下，OSC2 引脚输出 CLKO 振荡信号，该信号是 OSC1 引脚上振荡信号的 4 分频，该频率等于指令周期的倒数。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出

CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

注 1: 当配置位 CCP2MX 置 1 时，对 CCP2 进行默认分配。
注 2: 当配置位 CCP2MX 清零时，对 CCP2 进行其他分配。

表 1-3: PIC18F44J10/45J10 I/O 引脚排列说明 (续)

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	34	32	I/O O I	ST — ST	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1 外部时钟输入。
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 ⁽²⁾	16	35	35	I/O I I/O	ST CMOS ST	数字 I/O。 Timer1 振荡器输入。 捕捉 2 输入 / 比较器 2 输出 / PWM2 输出。
RC2/CCP1/P1A RC2 CCP1 P1A	17	36	36	I/O I/O O	ST ST —	数字 I/O。 捕捉 1 输入 / 比较器 1 输出 / PWM1 输出。 增强型 CCP1 输出。
RC3/SCK1/SCL1 RC3 SCK1 SCL1	18	37	37	I/O I/O I/O	ST ST ST	数字 I/O。 SPI 模式的同步串行时钟输入 / 输出。 I ² C™ 模式的同步串行时钟输入 / 输出。
RC4/SDI1/SDA1 RC4 SDI1 SDA1	23	42	42	I/O I I/O	ST ST ST	数字 I/O。 SPI 数据输入。 I ² C 数据 I/O。
RC5/SDO1 RC5 SDO1	24	43	43	I/O O	ST —	数字 I/O。 SPI 数据输出。
RC6/TX/CK RC6 TX CK	25	44	44	I/O O I/O	ST — ST	数字 I/O。 EUSART 异步发送。 EUSART 同步时钟 (参见相关 RX/DT 引脚)。
RC7/RX/DT RC7 RX DT	26	1	1	I/O I I/O	ST ST ST	数字 I/O。 EUSART 异步接收。 EUSART 同步数据 (参见相关 TX/CK 引脚)。

图注: TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

O = 输出

CMOS = CMOS 兼容输入或输出

I = 输入

P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。

2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

PIC18F45J10 系列

表 1-3: PIC18F44J10/45J10 I/O 引脚排列说明 (续)

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
RD0/PSP0/SCK2/ SCL2 RD0 PSP0 SCK2 SCL2	19	38	38	I/O I/O I/O I/O	ST TTL ST ST	PORTD 是双向 I/O 端口或与微处理器连接的并行从动端口 (Parallel Slave Port, PSP)。当使能 PSP 模块时, 这些引脚具有 TTL 输入缓冲器。 数字 I/O。 并行从动端口数据。 SPI 模式的同步串行时钟输入 / 输出。 I ² C™ 模式的同步串行时钟输入 / 输出。
RD1/PSP1/SDI2/SDA2 RD1 PSP1 SDI2 SDA2	20	39	39	I/O I/O I I/O	ST TTL ST ST	数字 I/O。 并行从动端口数据。 SPI 数据输入。 I ² C 数据 I/O。
RD2/PSP2/SDO2 RD2 PSP2 SDO2	21	40	40	I/O I/O O	ST TTL —	数字 I/O。 并行从动端口数据。 SPI 数据输出。
RD3/PSP3/SS2 RD3 PSP3 SS2	22	41	41	I/O I/O I	ST TTL TTL	数字 I/O。 并行从动端口数据。 SPI 从动选择输入。
RD4/PSP4 RD4 PSP4	27	2	2	I/O I/O	ST TTL	数字 I/O。 并行从动端口数据。
RD5/PSP5/P1B RD5 PSP5 P1B	28	3	3	I/O I/O O	ST TTL —	数字 I/O。 并行从动端口数据。 增强型 CCP1 输出。
RD6/PSP6/P1C RD6 PSP6 P1C	29	4	4	I/O I/O O	ST TTL —	数字 I/O。 并行从动端口数据。 增强型 CCP1 输出。
RD7/PSP7/P1D RD7 PSP7 P1D	30	5	5	I/O I/O O	ST TTL —	数字 I/O。 并行从动端口数据。 增强型 CCP1 输出。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。
注 2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

表 1-3: PIC18F44J10/45J10 I/O 引脚排列说明 (续)

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
$\overline{\text{RE0}}/\text{RD}/\text{AN5}$ $\overline{\text{RE0}}$ RD AN5	8	25	25	I/O I I	ST TTL 模拟	PORTE 是双向 I/O 端口。 数字 I/O。 并行从动端口读控制。 (参见 WR 和 CS 引脚)。 模拟输入 5。
$\overline{\text{RE1}}/\text{WR}/\text{AN6}$ $\overline{\text{RE1}}$ WR AN6	9	26	26	I/O I I	ST TTL 模拟	数字 I/O。 并行从动端口写控制 (参见 CS 和 RD 引脚)。 模拟输入 6。
$\overline{\text{RE2}}/\text{CS}/\text{AN7}$ $\overline{\text{RE2}}$ CS AN7	10	27	27	I/O I I	ST TTL 模拟	数字 I/O。 并行从动端口片选控制。 (参见相关的 RD 和 WR 引脚)。 模拟输入 7。
VSS	12, 31	6, 30, 31	6, 29	P	—	逻辑电路和 I/O 引脚的参考地。
VDD	11, 32	7, 8, 28, 29	7, 28	P	—	逻辑电路和 I/O 引脚的正电源。
$\text{VDDCORE}/\text{VCAP}$ VDDCORE VCAP	6	23	23	P P	— —	逻辑电路和 I/O 引脚的正电源。 逻辑电路和 I/O 引脚的参考地。
NC	—	13	12, 13, 33, 34	—	—	不连接。

图注: TTL = TTL 兼容输入
 ST = CMOS 电平的施密特触发器输入
 O = 输出
 CMOS = CMOS 兼容输入或输出
 I = 输入
 P = 电源

注 1: 当配置位 CCP2MX 置 1 时, 对 CCP2 进行默认分配。
 2: 当配置位 CCP2MX 清零时, 对 CCP2 进行其他分配。

PIC18F45J10 系列

注:

2.0 振荡器配置

2.1 振荡器类型

PIC18F45J10 系列器件可以在五种不同的振荡器模式下工作：

1. HS 高速晶振 / 谐振器
2. HSPLL 带软件 PLL 控制的高速晶振 / 谐振器
3. EC 带 Fosc/4 输出的外部时钟
4. ECPLL 带软件 PLL 控制的外部时钟
5. INTRC 31 kHz 内部振荡器

用户可以通过编程FOSC2:FOSC0配置位来选择这其中的前四种模式。而第五种模式（INTRC）可能要在软件控制下进入；也可以将它配置为器件复位时的默认模式。

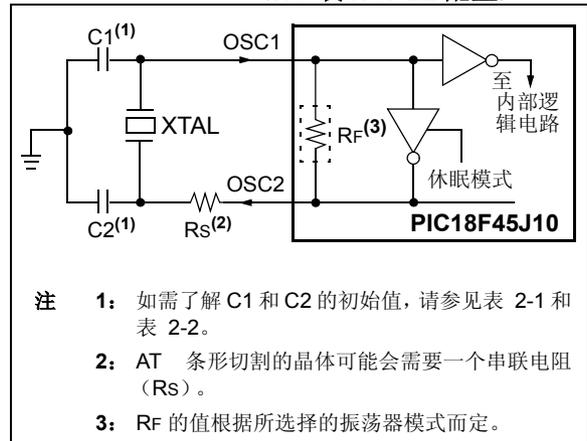
2.2 晶振 / 陶瓷谐振器（HS 模式）

在 HS 或 HSPLL 振荡器模式中，晶振或陶瓷谐振器与 OSC1 和 OSC2 引脚连接来产生振荡。图 2-1 显示了引脚连接方式。

振荡器的设计要求使用平行切割的晶体。

注： 使用顺序切割的晶体，会使振荡器产生的频率不在晶体制造厂商所给的参数范围内。

图 2-1: 晶振 / 陶瓷谐振器工作原理 (HS 或 HSPLL 配置)



- 注**
- 1: 如需了解 C1 和 C2 的初始值，请参见表 2-1 和表 2-2。
 - 2: AT 条形切割的晶体可能会需要一个串联电阻 (Rs)。
 - 3: Rf 的值根据所选择的振荡器模式而定。

表 2-1: 陶瓷谐振器的电容选择

使用的典型电容值:			
模式	频率	OSC1	OSC2
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

上述电容值仅供设计参考。
已在下列谐振器的基本起振和工作过程中测试了这些电容。**这些值不是最佳值。**
要得到合适的振荡器工作状态，可能需要不同的电容值。用户应在应用的预期 VDD 和温度范围内测试振荡器的性能。
欲知更多信息，请参见表 2-2 后的“注”。

所使用的谐振器:
4.0 MHz
8.0 MHz
16.0 MHz

PIC18F45J10 系列

表 2-2: 晶振的电容选择

振荡器类型	晶振频率	已测试的典型电容值:	
		C1	C2
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

上述电容值仅供设计参考。
 已在下列晶振的基本起振和运行过程中测试了这些电容。这些值不是最佳值。
 要得到合适的振荡器工作状况，可能需要不同的电容值。用户应在应用的预期 VDD 和温度范围内测试振荡器的性能。
 欲知更多信息，请参见本表后的“注”。

所使用的晶振:
4 MHz
8 MHz
20 MHz

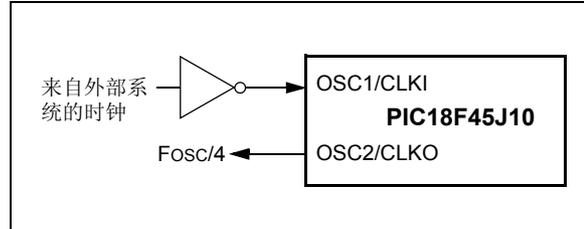
- 注 1:** 较高的电容值可以增加振荡器的稳定性，但同时也会增加起振时间。
- 注 2:** 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商询问外部元件的相应值。
- 注 3:** 为避免对低驱动电平规格的晶体造成过驱动，可能会需要使用电阻 R_s 。
- 注 4:** 请在应用中的预期 VDD 和温度范围内验证振荡器的性能。

2.3 外部时钟输入（EC 模式）

EC 和 ECPLL 振荡器模式需要在 OSC1 引脚连接一个外部时钟源。在上电复位后或从休眠模式退出后，不需要振荡器起振时间。

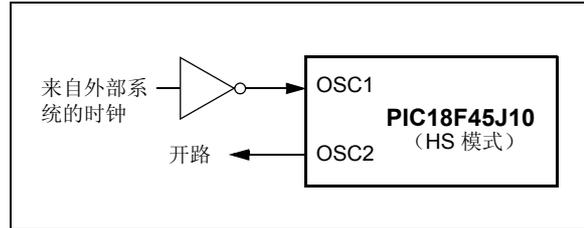
在 EC 振荡器模式下，振荡器频率的 4 分频信号可由 OSC2 引脚输出。此信号可用于测试或同步其他逻辑。图 2-2 显示了 EC 振荡器模式的引脚连接方式。

图 2-2: 外部时钟输入工作原理（EC 配置）



如图 2-3 所示，在 HS 模式下，OSC1 引脚也可以连接外部时钟源。在此配置中，OSC2 引脚上的 4 分频输出不可用。

图 2-3: 外部时钟输入工作原理（HS 振荡器配置）

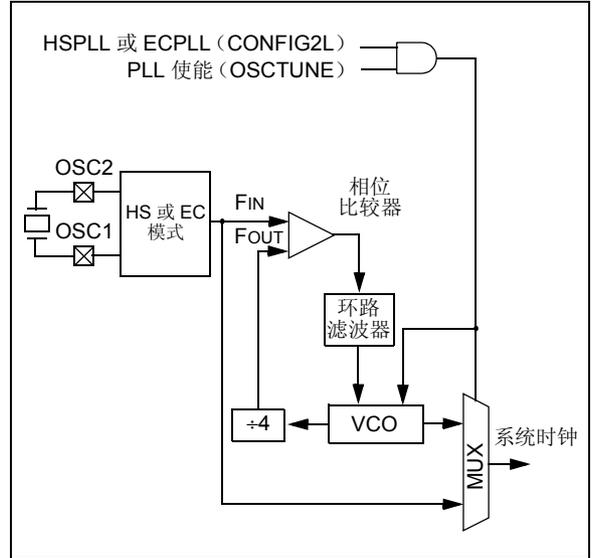


2.4 PLL 倍频器

如果用户希望使用低频晶振电路或通过晶振将器件频率调节至其最高额定频率，可以选择使用锁相环（Phase Locked Loop, PLL）电路。对于担心高频晶振引起 EMI 或需要内部振荡器提供高速时钟的用户而言，这样做可能会有用。由于这些原因，提供了 HSPLL 和 ECPLL 模式。

HSPLL 和 ECPLL 模式使器件能够有选择性地以外部振荡源的 4 倍速运行以产生最高为 40 MHz 的频率。通过在 OSCTUNE 寄存器（寄存器 2-1）中将 PLEN 位置 1 来使能 PLL。

图 2-4: PLL 框图



寄存器 2-1:

OSCTUNE: PLL 控制寄存器

U-0	R/W-0 ⁽¹⁾	U-0						
—	PLEN ⁽¹⁾	—	—	—	—	—	—	—
bit 7								bit 0

bit 7 未用：读为 0

bit 6 **PLEN**: 倍频器 PLL 使能位 ⁽¹⁾

1 = PLL 已使能

0 = PLL 已禁止

注 1: 用于 ECPLL 和 HSPLL 振荡器配置，否则，此位不可用并读为 0。

bit 5-0 未用：读为 0

图注:

R = 可读位

W = 可写位

U = 未用位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

2.5 内部振荡器电路

PIC18F45J10 系列器件包含内部振荡源 (INTRC)，它提供标称频率为 31 kHz 的输出。INTRC 在器件上电时使能并在器件配置周期期间作为器件时钟源，直至器件进入工作模式。如果选择 INTRC 作为器件时钟源或者使能了以下各项之一也会使能 INTRC：

- 故障保护时钟监视器
- 看门狗定时器
- 双速启动

在第 20.0 节“CPU 的特殊性能”中对这些功能进行了更详细的讨论。

也可以通过将 FOSC2 配置位置 1 把 INTRC 配置为器件启动时的默认时钟源。这将在第 2.6.1 节“振荡器控制寄存器”中进行讨论。

2.6 时钟源和振荡器切换

PIC18F45J10 系列提供的功能包括允许将器件时钟源从主振荡器切换到备用时钟源。PIC18F45J10 系列器件提供了两种备用时钟源。当备用时钟源使能时，各种功耗管理工作模式都可用。

基本上，这些器件有三种时钟源：

- 主振荡器
- 辅助振荡器
- 内部振荡器

主振荡器包括外部晶振和谐振器模式以及外部时钟模式。特定的模式由 FOSC2:FOSC0 配置位定义。这些模式的具体情况已在本章前面的内容中作过介绍。

辅助振荡器是不与 OSC1 或 OSC2 引脚连接的外部时钟源。这些时钟源即使在控制器处于功耗管理模式时仍然可以继续工作。

PIC18F45J10 系列器件提供 Timer1 振荡器作为辅助振荡器。此振荡器在所有功耗管理模式中通常是实时时钟等功能的时基。

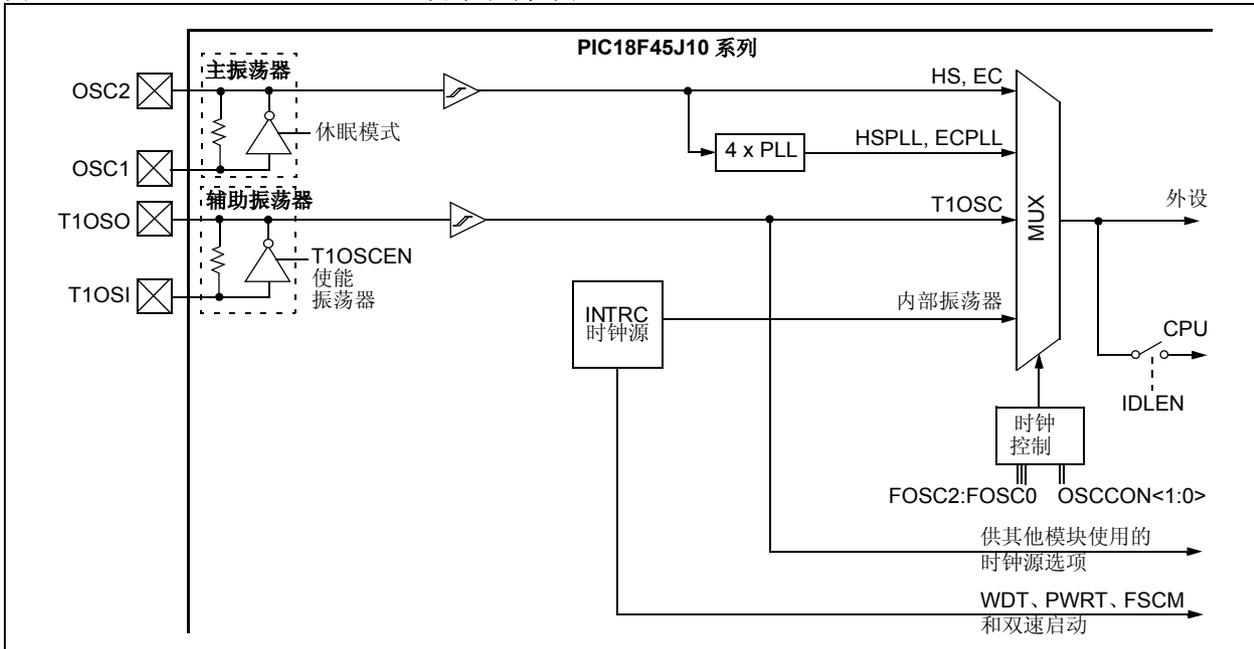
大部分情况下，在 RC0/T1OSO/T13CKI 和 RC1/T1OSI 引脚之间接有一个 32.768 kHz 的时钟晶振。在每个引脚与地之间同时接有负载电容。

第 11.3 节“Timer1 振荡器”中将对 Timer1 振荡器作更详细的讨论。

除了作为主时钟源之外，**内部振荡器**还可以作为功耗管理模式的时钟源。INTRC 源也可以作为几种特殊功能部件的时钟源，例如 WDT 和故障保护时钟监视器。

图 2-5 中所示为 PIC18F45J10 系列器件的时钟源。如需了解配置寄存器的详细信息，请参见第 20.0 节“CPU 的特殊性能”。

图 2-5: PIC18F45J10 系列时钟框图



2.6.1 振荡器控制寄存器

OSCCON 寄存器（寄存器 2-2）控制全功耗模式和功耗管理模式下器件时钟工作的几个方面。

系统时钟选择位 SCS1:SCS0 用于选择时钟源。可用的时钟源有主时钟（由 FOSC2:FOSC0 配置位定义）、辅助时钟（Timer1 振荡器）和内部振荡器。在写入一个或多个位后，有一段短的时钟切换间隔，之后，时钟源发生改变。

OSTS（OSCCON<3>）和 T1RUN（T1CON<6>）位表明当前由哪个时钟源提供器件时钟。OSTS 位置 1 表明振荡器起振定时器（OST）延时已结束，且主时钟在主时钟模式下提供器件时钟。T1RUN 位置 1 表明 Timer1 振荡器在辅助时钟模式下提供器件时钟。在功耗管理模式下的任何时间，这两个位中都只有一个位会被置 1。如果这两个位都没有置 1，则当前器件时钟源是 INTRC，或者内部振荡器刚刚起振且尚未稳定。

IDLEN 位决定当执行 SLEEP 指令时，器件是进入休眠模式还是某种空闲模式。

第 3.0 节“功耗管理模式”中将更详细地讨论 OSCCON 寄存器中的标志位和控制位的使用。

- | |
|---|
| <p>注 1: 要选择辅助时钟源，必须使能 Timer1 振荡器。通过将 Timer1 控制寄存器中的 T1OSCEN 位（T1CON<3>）置 1，可以使能 Timer1 振荡器。如果未使能 Timer1 振荡器，则在执行 SLEEP 指令期间任何选择辅助时钟源的操作都会被忽略。</p> <p>2: 建议当 Timer1 振荡器稳定工作之后再执行 SLEEP 指令，否则当 Timer1 振荡器起振时可能会发生很长的延迟。</p> |
|---|

2.6.1.1 系统时钟选择和 FOSC2 配置位

发生所有形式的复位时，SCS 位都会被清零。在器件的默认配置中，这意味着将 FOSC1:FOSC0（也就是 HS 或 EC 模式的一种）定义的主振荡器用作器件复位时的主时钟源。

复位时的默认时钟配置可以随着 FOSC2 配置位的改变而改变。此位的作用是在 SCS1:SCS0 = 00 时选择时钟源。当 FOSC2 = 1（默认）时，不管何时 SCS1:SCS0 = 00，都选择由 FOSC1:FOSC0 定义的振荡源。当 FOSC2 = 0 时，不管何时 SCS1:SCS2 = 00，都选择 INTRC 振荡器作为时钟源。因为在复位时 SCS 位被清零，所以 FOSC2 的设置也会更改复位时的默认振荡器模式。

不管 FOSC2 的设置如何，INTRC 总是会在器件上电时被使能。它将作为时钟源直到器件已经从存储器中装入了它的配置值为止。此时 FOSC 配置位被读取并选择了振荡器的工作模式。

注意主时钟或内部振荡器在任何给定时间都会有两种位设置选项，取决于 FOSC2 的设置。

2.6.2 振荡器转换

PIC18F45J10 系列器件包含了防止在切换时钟源时发生时钟“毛刺”的电路。在时钟切换时，器件时钟会有短暂的停顿。停顿的长度是旧时钟源的两个周期加上新时钟源的三到四个周期的和。此公式假设新时钟源是稳定的。

第 3.1.2 节“进入功耗管理模式”中对时钟转换进行了更详细的讨论。

PIC18F45J10 系列

寄存器 2-2:

OSCCON: 振荡器控制寄存器

R/W-0	U-0	U-0	U-0	R-q ⁽¹⁾	U-0	R/W-0	R/W-0
IDLEN	—	—	—	OSTS	—	SCS1	SCS0
bit 7				bit 0			

bit 7 **IDLEN:** 空闲使能位

- 1 = 在执行 SLEEP 指令时, 器件进入空闲模式
- 0 = 在执行 SLEEP 指令时, 器件进入休眠模式

bit 6-4 未用: 读为 0

bit 3 **OSTS:** 振荡器起振延时状态位⁽¹⁾

- 1 = 振荡器起振定时器延时已经结束; 主振荡器正在运行
- 0 = 振荡器起振定时器正在进行延时; 主振荡器尚未准备就绪

注 1: 当 HS 模式和双速启动都使能时, 复位值为 0; 否则, 复位值为 1。

bit 2 未用: 读为 0

bit 1-0 **SCS1:SCS0:** 系统时钟选择位

- 11 = 内部振荡器
- 10 = 主振荡器
- 01 = Timer1 振荡器

当 FOSC2 = 1 时:

- 00 = 主振荡器

当 FOSC2 = 0 时:

- 00 = 内部振荡器

图注:

U = 未用位, 读为 0

q = 值由配置确定

-n = 上电复位时的值 R = 可读位

0 = 清零

W = 可写位

2.7 功耗管理模式对各种时钟源的影响

当选择了 PRI_IDLE 模式时, 指定的主振荡器会继续运行而不中断。对于所有其他功耗管理模式, 使用 OSC1 引脚的振荡器会被禁止。OSC1 引脚 (以及由振荡器使用 OSC2 引脚) 将会停止振荡。

在辅助时钟模式下 (SEC_RUN 和 SEC_IDLE), Timer1 振荡器作为器件时钟源工作。如果需要, Timer1 振荡器也可以运行在所有功耗管理模式下为 Timer1 或 Timer3 提供时钟源。

在 RC_RUN 和 RC_IDLE 模式下, 内部振荡器提供器件时钟。无论在何种功耗管理模式下, 都可以直接使用 31 kHz 的 INTRC 输出来提供时钟或者使能它来支持多种特殊功能部件 (欲知更多有关 WDT、故障保护时钟监视器和双速启动的信息, 请参见第 20.2 节“看门狗定时器 (WDT)”到第 20.5 节“故障保护时钟监视器”)。

如果选择了休眠模式, 所有的时钟源都会停止。因为消除了所有的晶体管切换电流, 休眠模式能实现最小的器件电流消耗 (仅泄漏电流)。

在休眠期间使能任何片上功能将会增加休眠时的电流消耗。需要 INTRC 来支持 WDT 工作。Timer1 振荡器可以用来为实时时钟提供时钟源。不需要器件时钟源的其他功能部件也可以工作 (即 MSSP 从动器件、PSP 和 INTn 引脚等)。第 23.2 节“直流规范: 掉电和供电电流”列出了会明显增加电流消耗的外设。

2.8 上电延时

有两个定时器控制上电延迟，这样大部分应用都无需外接复位电路。上电延迟可以确保在器件电源稳定（常规环境下）且主时钟稳定工作之前器件保持复位状态。欲知有关上电延迟的其他信息，请参见第 4.5 节“上电延迟定时器（PWRT）”。

第一个定时器是上电延时定时器（PWRT），它在上电时提供一个固定的延迟（表 23-10 中的参数 33）。它总是使能的。

第二个定时器是振荡器起振定时器（OST），旨在使芯片在晶振稳定之前保持在复位状态（HS 模式）。OST 通过计数 1024 个振荡周期实现此延迟，并在延迟后允许振荡器为器件提供时钟。

在上电复位后，会有一段延迟间隔 TCSD（表 23-10 中的参数 38），控制器在这段时间中为执行指令做准备。

表 2-3: 休眠模式下 OSC1 和 OSC2 引脚状态

振荡器模式	OSC1 引脚	OSC2 引脚
EC, ECPLL	悬空，由外部时钟拉高	处于逻辑低电平（输出时钟的 4 分频信号）
HS, HSPLL	处于静态电平时，反馈反相器被禁止	处于静态电平时，反馈反相器被禁止

注：如需了解有关由于休眠和 MCLR 复位引起的延时的信息，请参见第 4.0 节“复位”中的表 4-2。

PIC18F45J10 系列

注:

3.0 功耗管理模式

PIC18F45J10 系列器件提供了只需通过管理 CPU 和外设的时钟源就可以管理功耗的功能。一般而言，降低时钟频率和减少由时钟源驱动的电路数目会使功耗降低。为了在应用中管理功耗，提供了三种主要的工作模式：

- 运行模式
- 空闲模式
- 休眠模式

这些模式定义了器件的哪些部分由时钟源驱动，以及以多高的时钟速度驱动。运行和空闲模式可以使用三种可用时钟源（主时钟源、辅助时钟源或内部振荡器电路）中的任意一种；而休眠模式则不使用时钟源。

功耗管理模式包括几种在以前的 PICmicro® 器件上提供的节省功耗的功能。其中之一就是在其他的 PIC18 器件上提供的时钟切换功能，允许控制器使用 Timer1 振荡器代替主振荡器。还包括所有 PICmicro 器件都提供的休眠模式，在此模式下器件时钟停止。

3.1 选择功耗管理模式

选择功耗管理模式需要考虑两个因素：是否用时钟源驱动 CPU 以及使用哪个时钟源。IDLEN 位 (OSCCON<7>) 控制 CPU 的时钟驱动，而 SCS1:SCS0 位 (OSCCON<1:0>) 选择时钟源。表 3-1 总结了各个模式、位设置、时钟源和受影响的模块。

3.1.1 时钟源

SCS1:SCS0 位可以为功耗管理模式选择三个时钟源中的一个。它们是：

- 主时钟，由 FOSC1:FOSC0 配置位定义
- 辅助时钟（Timer1 振荡器）
- 内部振荡器

3.1.2 进入功耗管理模式

从一种功耗管理模式切换到另一种功耗管理模式是通过装载 OSCCON 寄存器开始的。SCS1:SCS0 位选择时钟源并确定使用哪一种运行模式或空闲模式。更改这些位将导致立即切换到新的时钟源（假定新的时钟源正在运行）。切换也可能会遇到时钟转换延迟。第 3.1.3 节“时钟切换和状态指示位”和其后的章节将会讨论这些问题。

执行 SLEEP 指令可以触发进入功耗管理空闲模式或休眠模式。最后实际进入哪个模式由 IDLEN 位的状态决定。

更改功耗管理模式并不总是要求设置所有这些位，这由当前模式和将要切换到的模式决定。可以通过在发出 SLEEP 指令之前，更改振荡器选择位或更改 IDLEN 位来进行模式转换。如果已经正确地配置了 IDLEN 位，则只需通过执行 SLEEP 指令就能切换到所需的模式。

表 3-1: 功耗管理模式

模式	OSCCON 位		模块时钟控制		可用时钟和振荡器源
	IDLEN<7> ⁽¹⁾	SCS1:SCS0<1:0>	CPU	外设	
休眠	0	N/A	关闭	关闭	无——所有时钟被禁止
PRI_RUN	N/A	10	被时钟源驱动	被时钟源驱动	主时钟——HS, EC；这是正常的全功耗运行模式
SEC_RUN	N/A	01	被时钟源驱动	被时钟源驱动	辅助时钟——Timer1 振荡器
RC_RUN	N/A	11	被时钟源驱动	被时钟源驱动	内部振荡器
PRI_IDLE	1	10	关闭	被时钟源驱动	主时钟——HS, EC
SEC_IDLE	1	01	关闭	被时钟源驱动	辅助时钟——Timer1 振荡器
RC_IDLE	1	11	关闭	被时钟源驱动	内部振荡器

注 1: 仅当执行 SLEEP 指令时，IDLEN 才反映出有效值。

PIC18F45J10 系列

3.1.3 时钟切换和状态指示位

时钟源切换的延迟时间长度是旧时钟源的 2 个周期与新时钟源的 3 到 4 个周期的和。此公式假设新时钟源是稳定的。

OSTS (OSCCON<3>) 和 T1RUN (T1CON<6>) 两个位表示当前的时钟源及其状态。通常，在给定功耗管理模式下，这两个位中只有一个将被置 1。当 OSTS 位被置 1 时，主时钟提供器件时钟。当 T1RUN 位被置 1 时，Timer1 振荡器提供器件时钟。如果这两个位都不置 1，INTRC 为器件提供时钟。

注： 执行 SLEEP 指令不一定会使器件进入休眠模式。该指令充当触发条件，根据 IDLEN 位的设置，使控制器进入休眠模式或某种空闲模式。

3.1.4 多条 SLEEP 命令

用 SLEEP 指令启动的功耗管理模式由执行这条指令时 IDLEN 位的设置决定。如果执行另一条 SLEEP 指令，器件将进入由那时 IDLEN 位指定的功耗管理模式。如果 IDLEN 位已更改，则器件将进入由新的设置指定的功耗管理模式。

3.2 运行模式

在运行模式中，内核和外设的时钟均有效。这些模式之间的差异在于时钟源的不同。

3.2.1 PRI_RUN 模式

PRI_RUN 模式是单片机的正常的全功耗工作模式。除非使能双速启动，这也是器件复位时的默认模式（详细信息请参见第 20.4 节“双速启动”）。在此模式下，OSTS 位置 1（参见第 2.6.1 节“振荡器控制寄存器”）。

3.2.2 SEC_RUN 模式

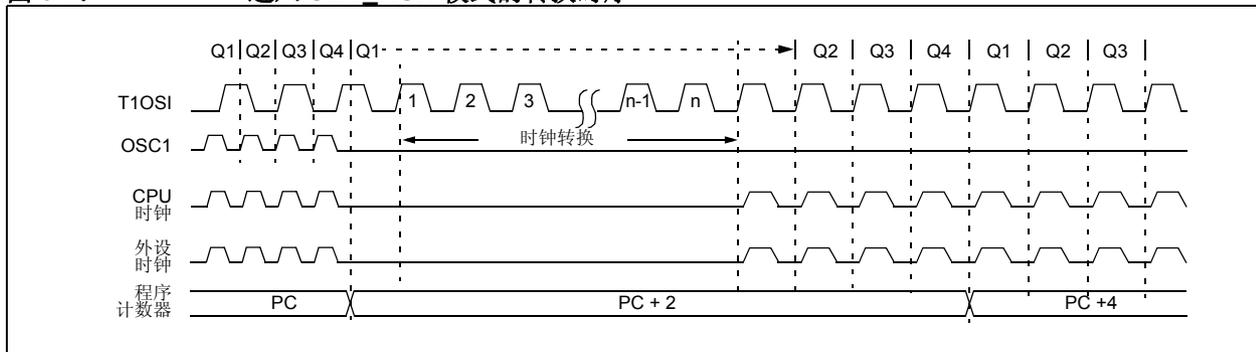
SEC_RUN 模式与其他 PIC18 器件提供的“时钟切换”功能兼容。在此模式下，CPU 和外设由 Timer1 振荡器提供时钟。这让用户能在仍使用高精度时钟源的情况下实现较低的功耗。

通过将 SCS1:SCS0 位设置为 01，器件进入 SEC_RUN 模式。器件时钟源切换到 Timer1 振荡器（见图 3-1），关闭主振荡器，T1RUN 位 (T1CON<6>) 置 1 并且 OSTS 位清零。

注： Timer1 振荡器应该在进入 SEC_RUN 模式之前已经开始运行。如果当 SCS1:SCS0 位被设置为“01”时，T1OSCEN 位未置 1，将不会进入 SEC_RUN 模式。如果 Timer1 振荡器已经被使能，但仍然没有开始运行，器件时钟将被延迟直到振荡器起振为止。在这种情况下，最初振荡器运行很不稳定，并且它的运行结果无法预料。

当从 SEC_RUN 模式转换到 PRI_RUN 模式时，在主时钟起振期间，外设和 CPU 继续使用 Timer1 振荡器作为时钟源。当主时钟准备就绪以后，时钟切换回主时钟（见图 3-2）。当时钟切换完成后，T1RUN 位被清零，OSTS 位被置 1，主时钟提供器件时钟。唤醒不会影响 IDLEN 和 SCS 位。Timer1 振荡器继续运行。

图 3-1: 进入 SEC_RUN 模式的转换时序



3.2.3 RC_RUN 模式

在 RC_RUN 模式下，使用内部振荡器作为 CPU 和外设的时钟源；主时钟关闭。在所有运行模式之中，此模式最节约功耗，且仍然执行代码。它非常适用于对时间精度要求不高或者不是总需要高速时钟的应用。

通过将 SCS 设置为“11”可以进入此模式。当时钟源切换到 INTRC（见图 3-2）时，主振荡器关闭，OSTS 位清零。

从 RC_RUN 模式切换到 PRI_RUN 模式时，在主时钟起振期间外设和器件继续使用 INTRC 作为时钟源。当主时钟准备就绪以后，时钟开始切换到主时钟（参见图 3-3）。当时钟切换完成后，OSTS 位被置 1，主时钟提供器件时钟。这种切换不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTRC 时钟源将继续运行。

图 3-2: 到 RC_RUN 模式的转换时序

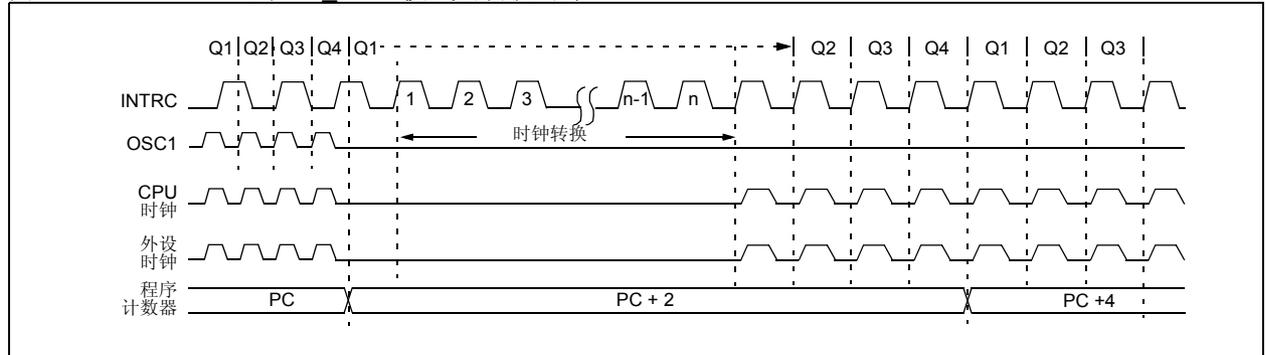
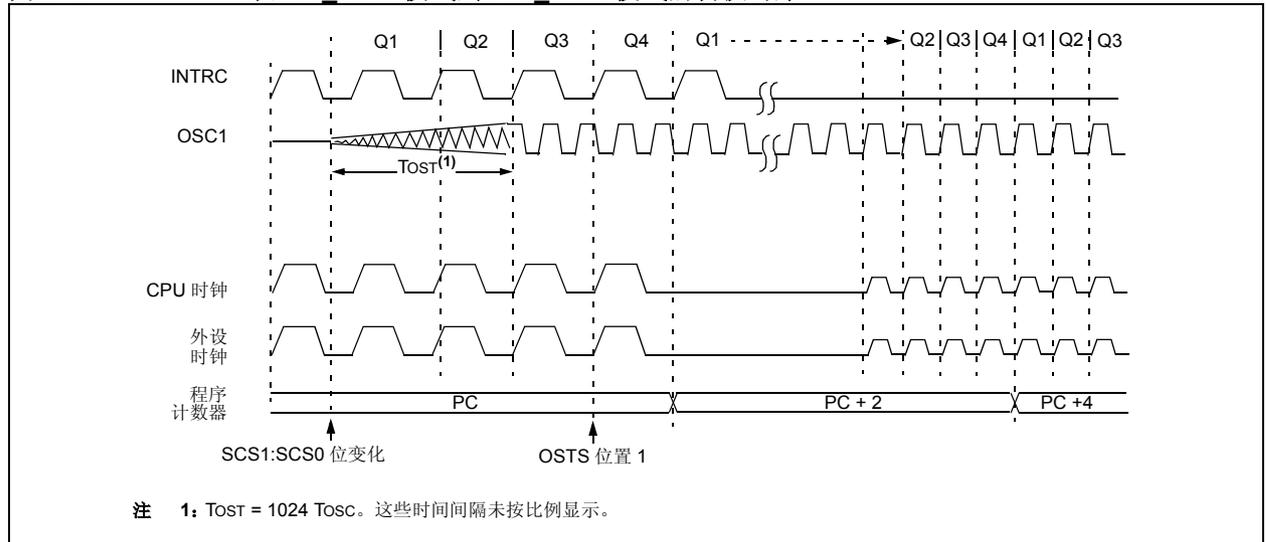


图 3-3: 从 RC_RUN 模式到 PRI_RUN 模式的转换时序



PIC18F45J10 系列

3.3 休眠模式

功耗管理休眠模式与所有其他 PICmicro 器件提供的传统休眠模式相同。通过清零 IDLEN 位（器件复位时的默认状态）并执行 SLEEP 指令进入该模式。这将关闭所选的振荡器（图 3-4）。并清零所有的时钟源状态位。

从其他模式进入休眠模式不需要时钟切换，这是因为一旦单片机进入休眠模式就不需要时钟了。如果选择了 WDT，INTRC 时钟源将继续运行。如果使能了 Timer1 振荡器，INTRC 时钟源也将继续运行。

在休眠模式下发生唤醒事件时（由于中断、复位或 WDT 超时），器件将没有时钟源直到由 SCS1:SCS0 位选定的时钟源准备就绪为止（见图 3-5），或者当双速启动或故障保护监视器被使能时，它会将内部振荡器作为时钟源（见第 20.0 节“CPU 的特殊性能”）。在这两种情况下，当主时钟提供器件时钟时，OSTS 位被置 1。唤醒不会影响 IDLEN 和 SCS 位。

3.4 空闲模式

空闲模式允许在外设继续运行的情况下，有选择地关闭单片机的 CPU。选择某种特定的空闲模式可使用户能进一步管理功耗。

执行 SLEEP 指令时，如果 IDLEN 位被置 1，外设将使用 SCS1:SCS0 位选定的时钟源；然而，将不会为 CPU 提供时钟。时钟源状态位不受影响。将 IDLEN 置 1 并执行 SLEEP 指令是一种从给定运行模式切换到其对应空闲模式的快速方法。

如果选择了 WDT，INTRC 时钟源将继续运行。如果使能了 Timer1 振荡器，INTRC 时钟源也将继续运行。

由于在空闲模式下 CPU 没有执行指令，只能通过中断、WDT 超时或复位使器件从空闲模式退出。当唤醒事件发生时，CPU 在准备好执行代码前，要延迟一个 T_{CSD} 的时间（表 23-10 中的参数 38）延迟。当 CPU 开始执行代码时，它将沿用与当前空闲模式相同的时钟源。例如，当从 RC_IDLE 模式唤醒时，将使用内部振荡器电路作为 CPU 和外设的时钟源（即 RC_RUN 模式）。唤醒不会影响 IDLEN 和 SCS 位。

当处于任何空闲模式或休眠模式时，WDT 超时将导致器件退出到由当前 SCS1:SCS0 位指定的运行模式。

图 3-4: 进入休眠模式的转换时序

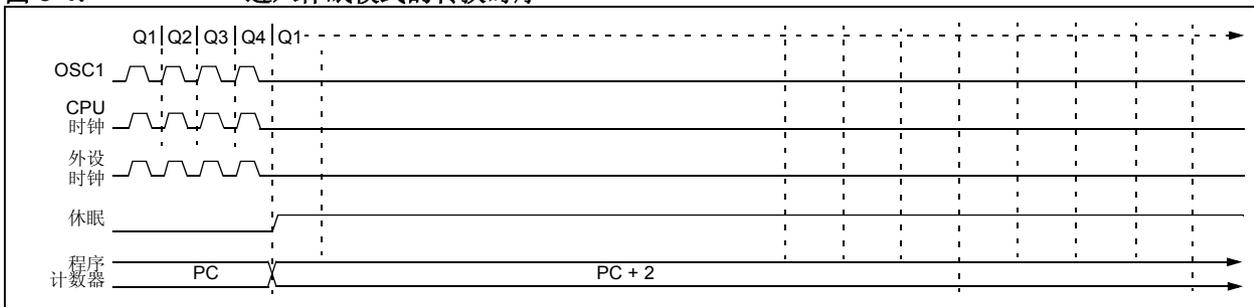
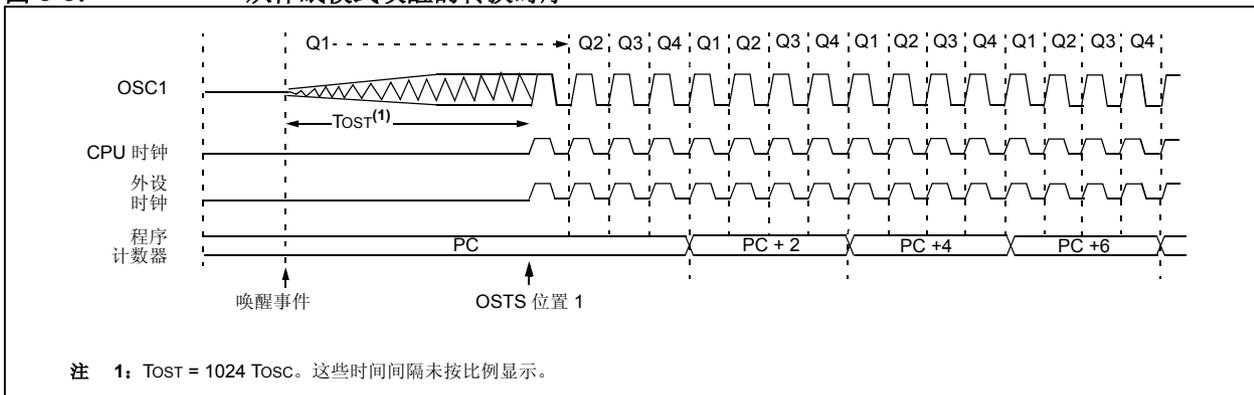


图 3-5: 从休眠模式唤醒的转换时序



注 1: T_{ost} = 1024 T_{osc}。这些时间间隔未按比例显示。

3.4.1 PRI_IDLE 模式

此模式在三种低功耗空闲模式中是唯一不禁止器件主时钟的。对于对时间精度要求很高的应用来说，由于时钟源不需要“热身”或从其他振荡器切换过来，选用此模式可以使用更加精确的主时钟源，并以最快的速度恢复器件运行。

通过置位 IDLEN 位并执行 SLEEP 指令可以从 PRI_RUN 模式进入 PRI_IDLE 模式。如果器件处于其他运行模式，请先置位 IDLEN，然后将 SCS 位置为“10”并执行 SLEEP 指令。虽然 CPU 被禁止，但外设仍继续使用由 FOSC0 控制位指定的主时钟源。OSTS 位保持置 1（见图 3-6）。

当唤醒事件发生时，由主时钟源为 CPU 提供时钟。在唤醒事件和代码执行开始之间需要一段 T_{CSD} 间隔的延迟。这段时间可以使 CPU 做好执行指令的准备。在唤醒之后，OSTS 位保持置 1。唤醒不会影响 IDLEN 和 SCS 位（见图 3-7）。

3.4.2 SEC_IDLE 模式

在 SEC_IDLE 模式中，CPU 被禁止，但外设继续使用 Timer1 振荡器作为时钟源。通过置位 IDLEN 位并执行 SLEEP 指令可以从 SEC_RUN 进入此模式。如果器件处于其他运行模式，请先置位 IDLEN，然后将 SCS1:SCS0 置为“01”并执行 SLEEP 指令。当时钟源切换到 Timer1 振荡器时，主振荡器关闭，OSTS 位清零，T1RUN 位被置 1。

当唤醒事件发生时，外设继续将 Timer1 振荡器作为时钟源。在唤醒事件后经过一个 T_{CSD} 的时间间隔，CPU 使用 Timer1 振荡器作为时钟源并开始执行代码。唤醒不会影响 IDLEN 和 SCS 位；Timer1 振荡器继续运行（见图 3-7）。

注： Timer1 振荡器应该在进入 SEC_IDLE 模式之前已经在运行了。如果执行 SLEEP 指令时 T1OSCCN 位没有被置 1，那么 SLEEP 指令会被忽略并且不会进入 SEC_IDLE 模式。如果 Timer1 振荡器已经被使能，但还没有开始运行，外设时钟将被延迟直到振荡器起振为止。在这种情况下，初始振荡器运行很不稳定并且它的运行结果无法预料。

图 3-6: 进入空闲模式的转换时序

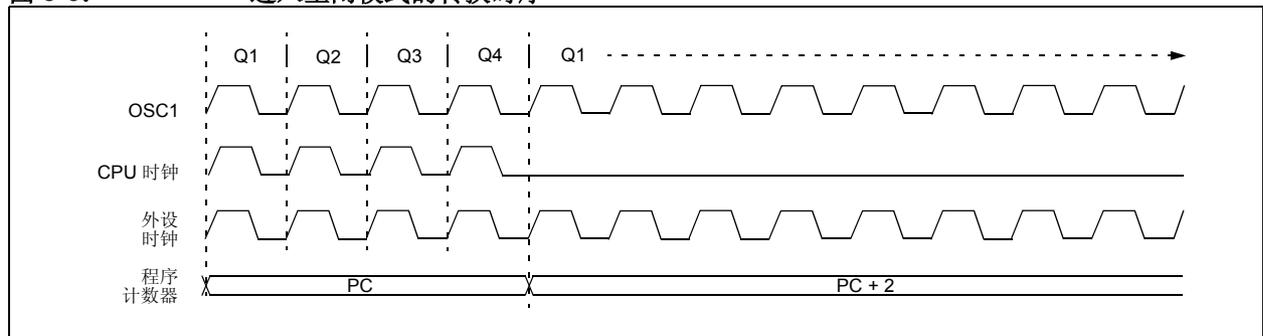
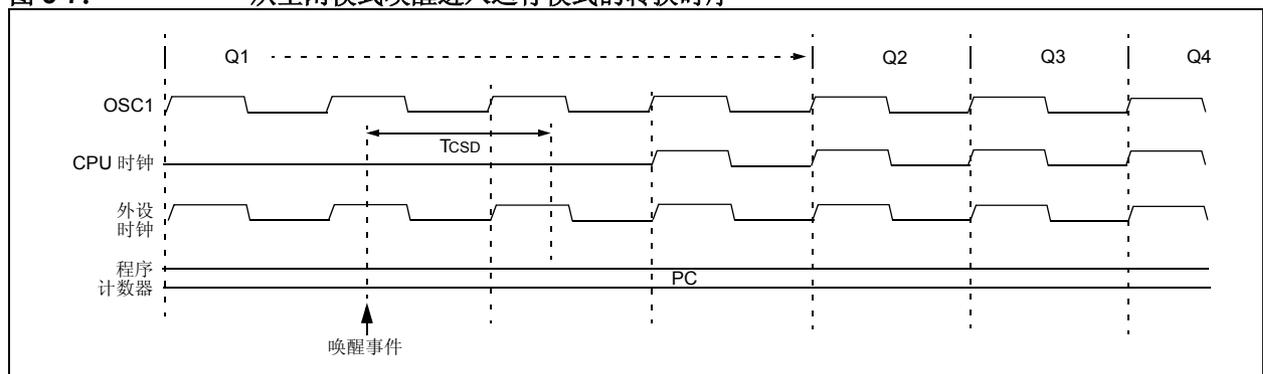


图 3-7: 从空闲模式唤醒进入运行模式的转换时序



3.4.3 RC_IDLE 模式

在 RC_IDLE 模式中，CPU 被禁止，但外设继续使用内部振荡器作为时钟源。该模式允许在器件处于空闲模式期间对功耗进行控制。

通过将 IDLEN 位置 1 并执行 SLEEP 指令可以从 RC_RUN 模式进入此模式。如果器件处于其他运行模式，请先置位 IDLEN，然后将 SCS 位清零并执行 SLEEP 指令。当时钟源切换到 INTRC 时，主振荡器关闭，OSTS 位被清零。

当唤醒事件发生时，外设继续使用 INTRC 作为时钟源。唤醒事件发生后经过一个 TcSD 的时间间隔，CPU 使用 INTRC 作为时钟源并开始执行代码。唤醒不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTRC 源将继续运行。

3.5 退出空闲和休眠模式

通过将中断、复位或 WDT 超时作为触发事件，从休眠模式或任何空闲模式退出。本节将讨论引起从功耗管理模式退出的触发事件。在每个功耗管理模式的章节中，还讨论了时钟控制子系统的操作（见第 3.2 节“运行模式”、第 3.3 节“休眠模式”和第 3.4 节“空闲模式”）。

3.5.1 通过中断退出

任何可用的中断源可以引起器件从空闲模式或休眠模式退出，并进入运行模式。要使能此功能，必须通过将 INTCON 或 PIE 寄存器中的相应中断允许位置 1 来使能该中断源。当相应的中断标志位被置 1 时，启动退出时序。

当通过中断从空闲或休眠模式退出时，如果 GIE/GIEH 位 (INTCON<7>) 置 1，程序就会跳转到中断矢量处执行。否则，代码执行就会继续或恢复，而不发生跳转（见第 8.0 节“中断”）。

通过中断从休眠或空闲模式退出时，一个固定的延迟时间间隔 TcSD 是必需的。CPU 需要此延迟来为执行代码做准备。在此延迟后的第一个时钟周期恢复指令执行。

3.5.2 通过 WDT 超时退出

根据 WDT 超时发生时器件所处的不同功耗管理模式，会引发不同的操作。

如果器件没有执行代码（在所有空闲模式和休眠模式下），超时将导致从功耗管理模式退出（见第 3.2 节“运行模式”和第 3.3 节“休眠模式”）。如果器件正在执行代码（在所有运行模式下），超时将导致 WDT 复位（见第 20.2 节“看门狗定时器 (WDT)”）。

WDT 定时器和后分频器通过以下任一事件清零：

- 执行 SLEEP 或 CLRWDT 指令
- 当前选定的时钟源失效（使能故障保护时钟监视器时）

3.5.3 通过复位退出

通过复位从空闲或休眠模式退出，将自动强制器件使用 INTRC 作为时钟源运行。

3.5.4 在没有振荡器起振延迟的情况下退出

某些从功耗管理模式退出的方式根本不会启动 OST。具体有两种情形：

- 主时钟源处于不会被停止的 PRI_IDLE 模式；
- 主时钟源处于 EC 模式。

在这两种情况下，主时钟源都不需要振荡器起振延迟，因为要么它已经在运行了（在 PRI_IDLE 模式下），要么根本不需要振荡器起振延迟（在 EC 模式下）。然而，当退出休眠和空闲模式时，在唤醒事件后的一个固定的延迟间隔 TcSD 仍然是必需的，CPU 利用这段时间为执行代码做准备。在此延迟后的第一个时钟周期恢复指令执行。

4.0 复位

PIC18F45J10 系列器件有以下几种不同的复位方式：

- 上电复位（Power-on Reset, POR）
- 正常工作状态下的 MCLR 复位
- 功耗管理模式下的 MCLR 复位
- 看门狗定时器（WDT）复位（执行程序期间）
- 欠压复位（Brown-on Reset, BOR）
- RESET 指令
- 堆栈满复位
- 堆栈下溢复位

本节将讨论由 MCLR、POR 和 BOR 产生的各种复位以及各种起振定时器的操作。堆栈复位事件将在第 5.1.2.4 节“堆栈满和下溢复位”中讨论。WDT 复位将在第 20.2 节“看门狗定时器（WDT）”中讨论。

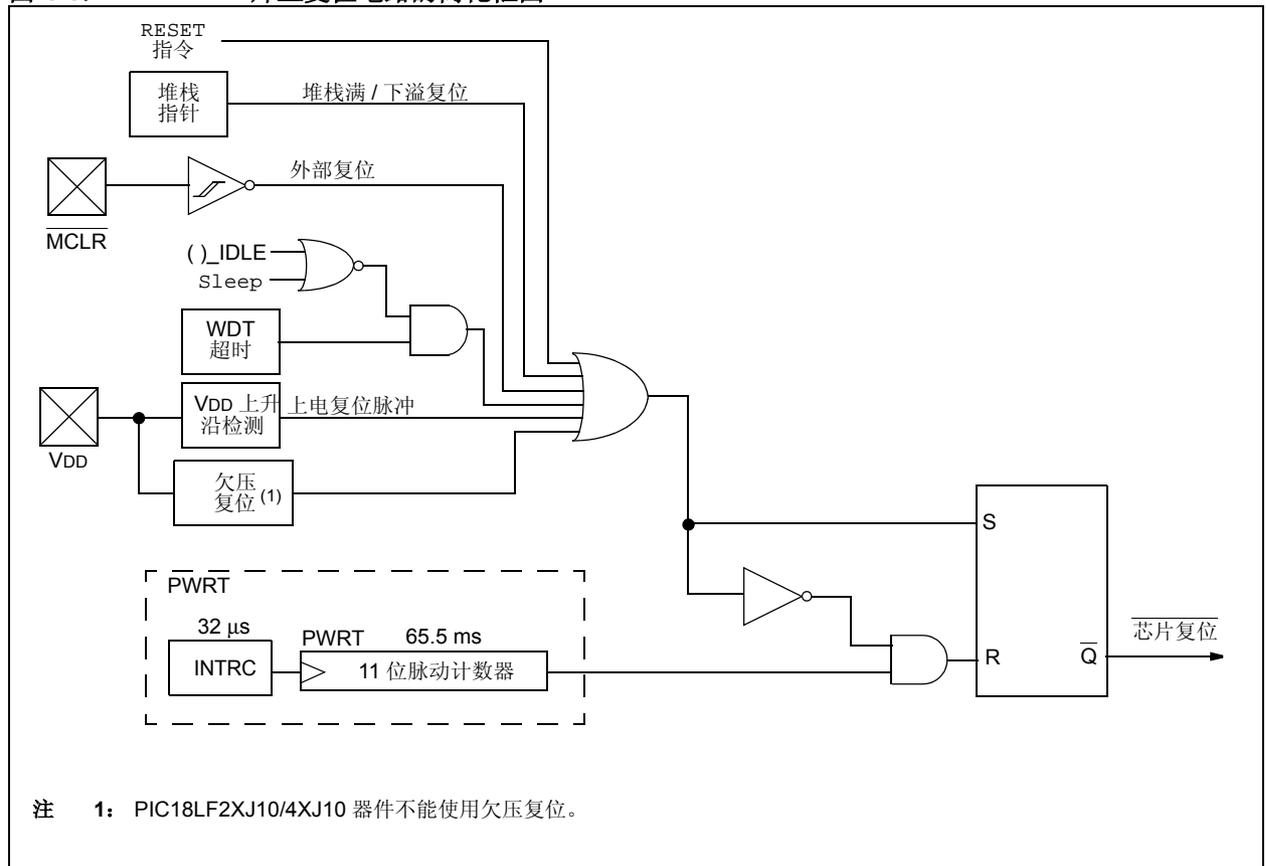
图 4-1 给出了片上复位电路的简化框图。

4.1 RCON 寄存器

可通过 RCON 寄存器（寄存器 4-1）跟踪器件复位事件。该寄存器的低 5 位表示特定的复位事件是否已经发生。在大部分情况下，只有复位事件可以将这些位置 1，而且它们必须在复位事件之后由应用程序清零。将这些标志位的状态一起读出可以得知刚发生的复位的类型。第 4.6 节“寄存器的复位状态”对此进行了更详细地说明。

RCON 寄存器还有一个设置中断优先级的控制位（IPEN）。中断优先级将在第 8.0 节“中断”中进行讨论。

图 4-1: 片上复位电路的简化框图



PIC18F45J10 系列

寄存器 4-1: **RCON: 复位控制寄存器**

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}
bit 7							bit 0

- bit 7 **IPEN:** 中断优先级使能位
 1 = 使能中断优先级
 0 = 禁止中断优先级 (PIC16CXXX 兼容模式)
- bit 6-5 **未用:** 读为 0
- bit 4 **\overline{RI} :** RESET 指令标志位
 1 = 未执行 RESET 指令 (仅由固件置 1)
 0 = 执行 RESET 指令导致器件复位 (必须在欠压复位发生之后由软件置 1)
- bit 3 **\overline{TO} :** 看门狗定时器超时标志位
 1 = 通过上电、CLRWDT 指令或 SLEEP 指令置 1
 0 = 发生了 WDT 超时
- bit 2 **\overline{PD} :** 掉电检测标志位
 1 = 通过上电或 CLRWDT 指令置 1
 0 = 通过执行 SLEEP 指令置 1
- bit 1 **\overline{POR} :** 上电复位状态位
 1 = 未发生上电复位 (仅由固件置 1)
 0 = 发生了上电复位 (必须在发生上电复位后由软件置 1)
- bit 0 **\overline{BOR} :** 欠压复位状态位
 1 = 未发生欠压复位 (仅由固件置 1)
 0 = 发生了欠压复位 (必须在欠压复位发生之后由软件置 1)

注: PIC18LF2XJ10/4XJ10 器件不能使用欠压复位。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

- 注 1: 建议在检测到上电复位后, 将 \overline{POR} 位置 1, 以便检测后续发生的上电复位。
- 注 2: 如果禁止了片上稳压器, \overline{BOR} 则总是保持为 0。欲知更多信息, 请参见第 4.4.1 节“检测 BOR”。
- 注 3: 欠压复位是指当 \overline{BOR} 为 0 且 \overline{POR} 为 1 时发生的复位 (假定在上电复位发生后 \overline{POR} 立即被软件置 1)。

4.2 主复位 ($\overline{\text{MCLR}}$)

$\overline{\text{MCLR}}$ 引脚提供了用外部硬件触发器件复位的方法。保持该引脚为低电平就能触发此复位。PIC18 扩展的单片机器件在 $\overline{\text{MCLR}}$ 复位信号传输路径中有一个噪声滤波器，它可以检测并滤除小的干扰脉冲。

任何内部复位，包括 WDT 复位，都不能将 $\overline{\text{MCLR}}$ 引脚驱动为低电平。

4.3 上电复位 (POR)

只要 VDD 上升到一定的门限以上，就会在片上产生上电复位信号。这使得器件在 VDD 上升到足够其工作时，以初始状态启动。

为了使用 POR 电路，可以将 $\overline{\text{MCLR}}$ 引脚通过一个电阻 ($1\text{ k}\Omega$ 到 $10\text{ k}\Omega$) 连接到 VDD。这样可以省去产生上电复位延时通常所需的外部 RC 元件。VDD 的最小上升速率已指定 (参数 D004)。对于延缓 VDD 上升时间的情况，请参见图 4-2。

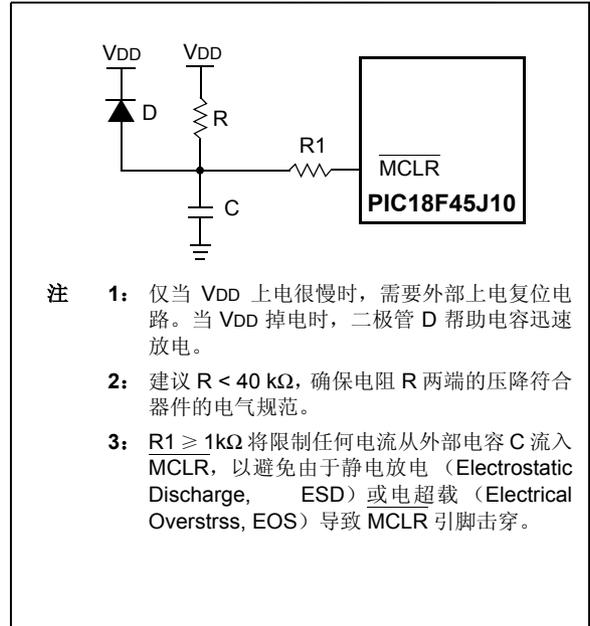
当器件开始正常工作 (即退出复位状态) 时，必须满足特定的工作参数要求 (电压、频率和温度等)，才能确保其正常工作。如果这些条件不满足，那么器件必须保持在复位状态，直到工作条件满足为止。

上电复位事件由 $\overline{\text{POR}}$ 位 (RCON<1>) 捕捉。只要发生上电复位，此位的状态就被置为 0，它不会因任何其他复位事件发生改变。任何硬件事件都不会将 $\overline{\text{POR}}$ 位复位为 1。要捕捉多个事件，用户必须在上电复位之后用软件手动将该位复位为 1。

4.4 欠压复位 (BOR) (仅 PIC18F2XJ10/4XJ10 器件)

一旦发生欠压复位，上电延时定时器就将芯片保持在复位状态达 TPWRT (参数 33) 时间。如果上电延迟定时器运行时，VDD 电压降到 VBOR 以下，芯片将重新回到欠压复位状态并初始化上电延迟定时器。一旦 VDD 电压上升到 VBOR 以上，上电延迟定时器将再执行一个延时。

图 4-2: 外部上电复位电路
(用于延缓 VDD 上电)



4.4.1 检测 BOR

$\overline{\text{BOR}}$ 位在发生欠压复位或上电复位事件时总是复位为 0。所以仅通过读 $\overline{\text{BOR}}$ 的状态很难判断 BOR 事件是否已经发生。更可靠的方法是同时检查 $\overline{\text{POR}}$ 和 $\overline{\text{BOR}}$ 的状态。假设在上电复位事件之后立即用软件将 $\overline{\text{POR}}$ 位复位为 1。如果 $\overline{\text{BOR}}$ 为 0 而 $\overline{\text{POR}}$ 为 1，就可以可靠地判断出已经发生了欠压复位事件。

器件编号中带有“LF”的器件 (如 PIC18LF25J10) 禁止欠压复位功能。在这种情况下，不能使用 $\overline{\text{BOR}}$ 位来确定欠压复位事件。上电复位事件仍然会清零 $\overline{\text{BOR}}$ 位。

PIC18F45J10 系列

4.5 上电延时定时器 (PWRT)

PIC18F45J10 系列器件加入了片上上电延时定时器 (PWRT) 以帮助稳定上电复位过程。PWRT 总是使能的。主要功能是确保在执行代码之前，器件的电压是稳定的。

PIC18F45J10 系列器件的上电延时定时器 (PWRT) 是一个 11 位计数器，使用 INTRC 时钟源作为时钟输入。这就产生大约 $2048 \times 32 \mu\text{s} = 65.6 \text{ ms}$ 的延迟时间。当 PWRT 计数时，器件保持在复位状态。

上电延迟时间取决于 INTRC 时钟，并且由于温度和工艺的变化，不同芯片的延迟时间各不相同。欲知详细信息，参见 DC 参数 33。

4.5.1 延时序列

如果使能 PWRT，在 POR 脉冲被清零后触发 PWRT 延时。总延时会根据 PWRT 的状态而有所不同。图 4-3、图 4-4、图 4-5 和图 4-6 都说明了在使能上电延时定时器时的上电延时序列。

由于延时是由上电复位脉冲引起的，因此若 $\overline{\text{MCLR}}$ 引脚保持足够长时间的低电平，延时将结束。将 $\overline{\text{MCLR}}$ 引脚的电平拉高后程序将立即执行 (图 4-5)。这对于测试或同步多个并行工作的 PIC18F 器件来说是非常有用的。

图 4-3: 上电延时序列 ($\overline{\text{MCLR}}$ 连接到 VDD, VDD 上升时间 < TPWRT)

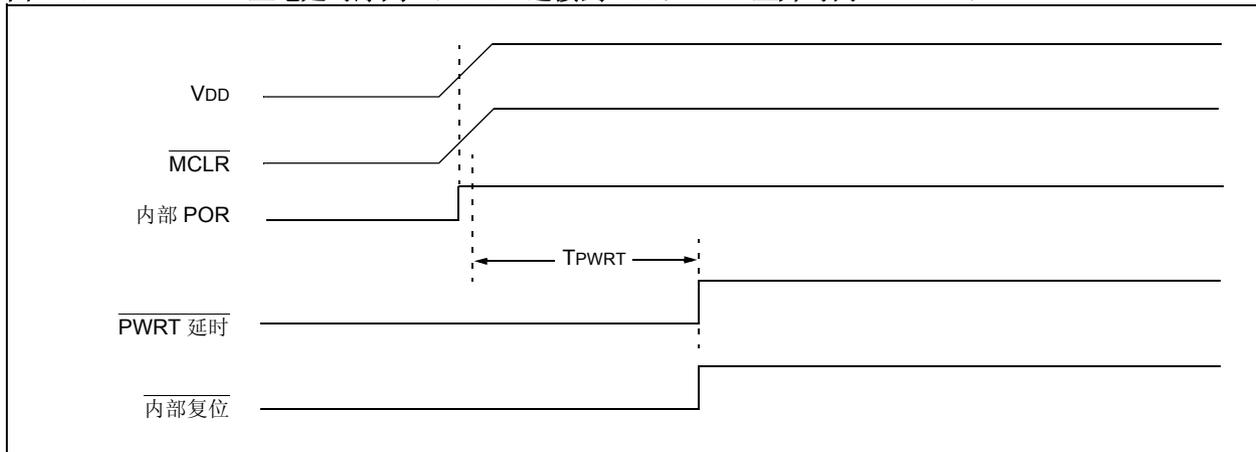


图 4-4: 上电延时序列 ($\overline{\text{MCLR}}$ 不连接到 VDD): 情形 1

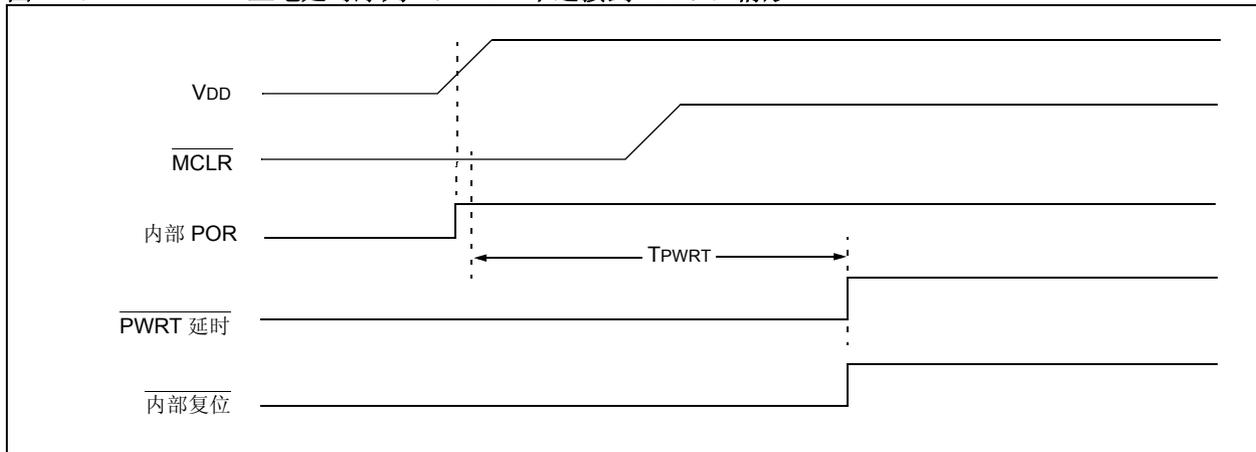


图 4-5: 上电延序列 ($\overline{\text{MCLR}}$ 不连接到 VDD): 情形 2

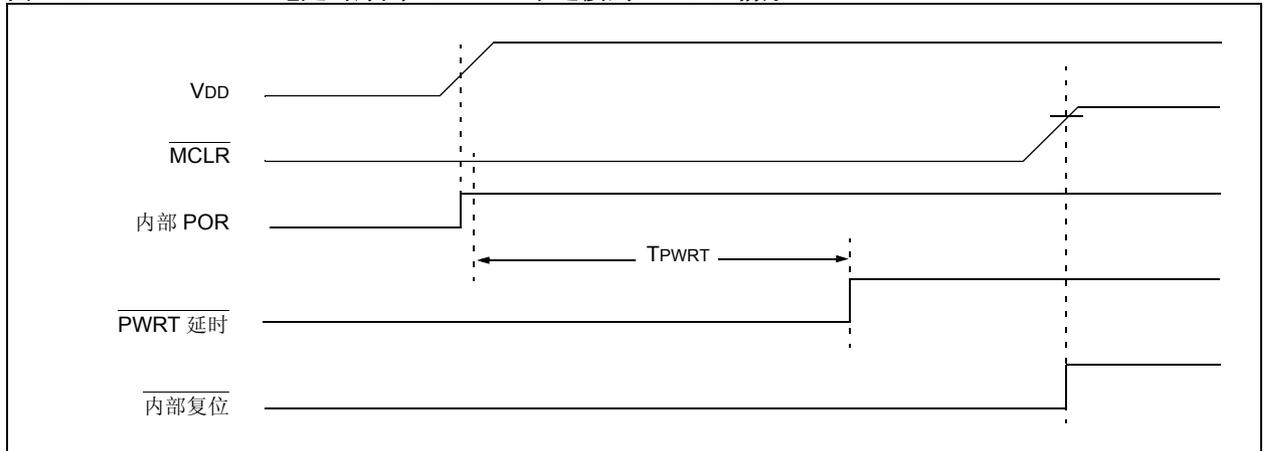
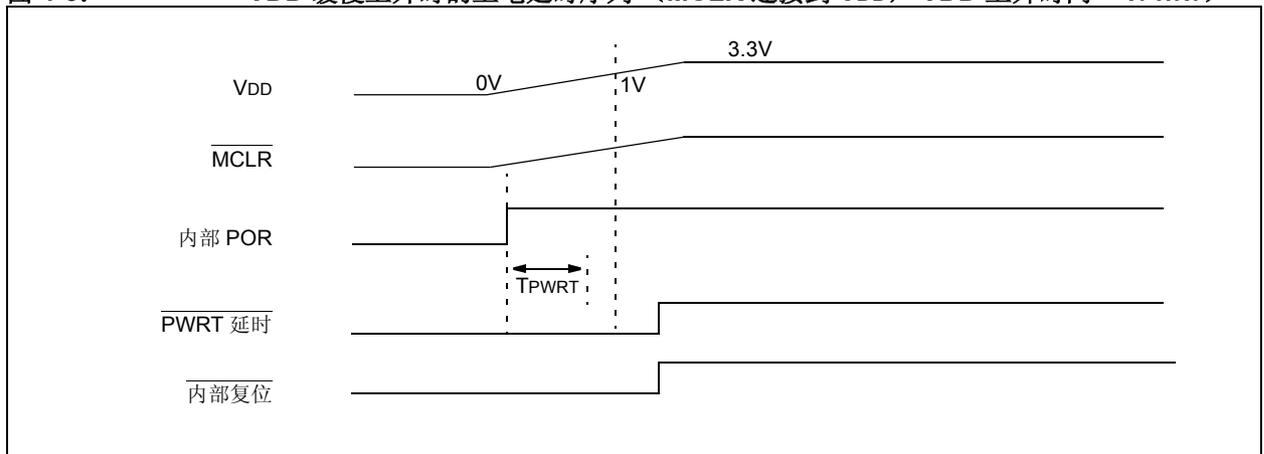


图 4-6: VDD 缓慢上升时的上电延序列 ($\overline{\text{MCLR}}$ 连接到 VDD, VDD 上升时间 > T_{PWRT})



PIC18F45J10 系列

4.6 寄存器的复位状态

大多数寄存器不受复位的影响。在上电复位时这些寄存器的状态未知，而在其他复位时它们的状态不变。而其他寄存器则根据不同的复位类型被强制置为相应的“复位状态”。

大多数寄存器不受 WDT 唤醒的影响，这是因为 WDT 唤醒被视为恢复正常工作。如表 4-1 所示，RCON 寄存器的状态位 RI、TO、PD、POR 和 BOR，在不同的复位情况下置 1 和清零的状态也各不相同。可在软件中使用这些状态位判断复位的性质。

表 4-2 说明了所有特殊功能寄存器的复位状态。这些复位被分为上电和欠压复位、主复位和 WDT 复位以及 WDT 唤醒复位。

表 4-1: RCON 寄存器的状态位、它们的含义以及初始化状态

条件	程序计数器 ⁽¹⁾	RCON 寄存器					STKPTR 寄存器	
		RI	TO	PD	POR	BOR ⁽²⁾	STKFUL	STKUNF
上电复位	0000h	1	1	1	0	0	0	0
RESET 指令	0000h	0	u	u	u	u	u	u
欠压	0000h	1	1	1	u	0	u	u
功耗管理运行模式期间的 MCLR	0000h	u	1	u	u	u	u	u
功耗管理空闲模式和休眠模式期间的 MCLR	0000h	u	1	0	u	u	u	u
全功耗或功耗管理运行模式期间的 WDT 超时	0000h	u	0	u	u	u	u	u
全功耗运行期间的 MCLR	0000h	u	u	u	u	u	u	u
堆栈满复位 (STVREN = 1)	0000h	u	u	u	u	u	1	u
堆栈下溢复位 (STVREN = 1)	0000h	u	u	u	u	u	u	1
堆栈下溢错误 (不是真正的复位, STVREN = 0)	0000h	u	u	u	u	u	u	1
功耗管理空闲或休眠模式期间的 WDT 超时	PC + 2	u	0	0	u	u	u	u
中断使器件从功耗管理模式退出	PC + 2	u	u	0	u	u	u	u

图注: u = 不变

注 1: 当芯片被中断唤醒且 GIEH 或 GIEL 位置 1 时, PC 装入中断矢量 (0008h 或 0018h)。

注 2: PIC18LF2XJ10/4XJ10 器件不能使用欠压复位。

表 4-2: 所有寄存器的初始化状态

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
TOSU	PIC18F2XJ10	PIC18F4XJ10	---0 0000	---0 0000	---0 uuuu ⁽¹⁾
TOSH	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
TOSL	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
STKPTR	PIC18F2XJ10	PIC18F4XJ10	00-0 0000	uu-0 0000	uu-u uuuu ⁽¹⁾
PCLATU	PIC18F2XJ10	PIC18F4XJ10	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	PC + 2 ⁽²⁾
TBLPTRU	PIC18F2XJ10	PIC18F4XJ10	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F2XJ10	PIC18F4XJ10	0000 000x	0000 000u	uuuu uuuu ⁽³⁾
INTCON2	PIC18F2XJ10	PIC18F4XJ10	1111 -1-1	1111 -1-1	uuuu -u-u ⁽³⁾
INTCON3	PIC18F2XJ10	PIC18F4XJ10	11-0 0-00	11-0 0-00	uu-u u-uu ⁽³⁾
INDF0	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTINC0	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTDEC0	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PREINC0	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PLUSW0	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
FSR0H	PIC18F2XJ10	PIC18F4XJ10	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTINC1	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTDEC1	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PREINC1	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PLUSW1	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
FSR1H	PIC18F2XJ10	PIC18F4XJ10	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F2XJ10	PIC18F4XJ10	---- 0000	---- 0000	---- uuuu

图注: u = 不变, x = 未知, - = 未用 (读为 0), q = 取值视具体情况而定。
阴影单元表示条件不适用于指定器件。

- 注 1:** 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2:** 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断矢量地址 (0008h 或 0018h)。
- 3:** INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (用以唤醒器件)。
- 4:** 关于特定条件下的复位值, 请参见表 4-1。

PIC18F45J10 系列

表 4-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
INDF2	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTINC2	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
POSTDEC2	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PREINC2	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
PLUSW2	PIC18F2XJ10	PIC18F4XJ10	N/A	N/A	N/A
FSR2H	PIC18F2XJ10	PIC18F4XJ10	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F2XJ10	PIC18F4XJ10	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F2XJ10	PIC18F4XJ10	0--- q-00	0--- q-00	u--- q-uu
WDTCON	PIC18F2XJ10	PIC18F4XJ10	---- --0	---- --0	---- --u
RCON ⁽⁴⁾	PIC18F2XJ10	PIC18F4XJ10	0--1 11q0	0--q qquu	u--u qquu
TMR1H	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	PIC18F2XJ10	PIC18F4XJ10	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
PR2	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	1111 1111
T2CON	PIC18F2XJ10	PIC18F4XJ10	-00 0000	-00 0000	-uuu uuuu
SSP1BUF	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP1STAT	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
ADRESH	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F2XJ10	PIC18F4XJ10	0-00 0000	0-00 0000	u-uu uuuu
ADCON1	PIC18F2XJ10	PIC18F4XJ10	--00 0qqq	--00 0qqq	--uu uqqq
ADCON2	PIC18F2XJ10	PIC18F4XJ10	0-00 0000	0-00 0000	u-uu uuuu

图注: u = 不变, x = 未知, - = 未用 (读为 0), q = 取值视具体情况而定。
阴影单元表示条件不适用于指定器件。

- 注 1: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断矢量地址 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (用以唤醒器件)。
- 4: 关于特定条件下的复位值, 请参见表 4-1。

表 4-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
CCPR1H	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F2XJ10	PIC18F4XJ10	--00 0000	--00 0000	--uu uuuu
BAUDCON	PIC18F2XJ10	PIC18F4XJ10	01-0 0-00	01-0 0-00	uu-u u-uu
ECCP1DEL	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
ECCP1AS	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
CVRCON	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
CMCON	PIC18F2XJ10	PIC18F4XJ10	0000 0111	0000 0111	uuuu uuuu
SPBRGH	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SPBRG	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
RCREG	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
TXREG	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA	PIC18F2XJ10	PIC18F4XJ10	0000 0010	0000 0010	uuuu uuuu
RCSTA	PIC18F2XJ10	PIC18F4XJ10	0000 000x	0000 000x	uuuu uuuu
EECON2	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
EECON1	PIC18F2XJ10	PIC18F4XJ10	---0 x00-	---0 x00-	---u uu-
IPR3	PIC18F2XJ10	PIC18F4XJ10	11-- ----	11-- ----	uu-- ----
PIR3	PIC18F2XJ10	PIC18F4XJ10	00-- ----	00-- ----	uu-- ---- ⁽³⁾
PIE3	PIC18F2XJ10	PIC18F4XJ10	00-- ----	00-- ----	uu-- ----
IPR2	PIC18F2XJ10	PIC18F4XJ10	11-- 1--1	11-- 1--1	uu-- u--u
PIR2	PIC18F2XJ10	PIC18F4XJ10	00-- 0--0	00-- 0--0	uu-- u--u ⁽³⁾
PIE2	PIC18F2XJ10	PIC18F4XJ10	00-- 0--0	00-- 0--0	uu-- u--u
IPR1	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE1	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu

图注: u = 不变, x = 未知, - = 未用 (读为 0), q = 取值视具体情况而定。
阴影单元表示条件不适用于指定器件。

- 注 1: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断矢量地址 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (用以唤醒器件)。
- 4: 关于特定条件下的复位值, 请参见表 4-1。

PIC18F45J10 系列

表 4-2: 所有寄存器的初始化状态(续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
TRISE	PIC18F2XJ10	PIC18F4XJ10	0000 -111	1111 -111	uuuu -uuu
TRISD	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F2XJ10	PIC18F4XJ10	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F2XJ10	PIC18F4XJ10	--1- 1111	--1- 1111	--u- uuuu
SSP2BUF	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATE	PIC18F2XJ10	PIC18F4XJ10	---- -xxx	---- -uuu	---- -uuu
LATD	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F2XJ10	PIC18F4XJ10	--x- xxxx	--u- uuuu	--u- uuuu
SSP2ADD	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP2STAT	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F2XJ10	PIC18F4XJ10	0000 0000	0000 0000	uuuu uuuu
PORTE	PIC18F2XJ10	PIC18F4XJ10	---- -xxx	---- -uuu	---- -uuu
PORTD	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F2XJ10	PIC18F4XJ10	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	PIC18F2XJ10	PIC18F4XJ10	--0- 0000	--0- 0000	--u- uuuu

图注: u = 不变, x = 未知, - = 未用 (读为 0), q = 取值视具体情况而定。
阴影单元表示条件不适用于指定器件。

- 注 1: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断矢量地址 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (用以唤醒器件)。
- 4: 关于特定条件下的复位值, 请参见表 4-1。

5.0 存储器构成

在 PIC18 增强型单片机器件上有两种类型的存储器：

- 程序存储器
- 数据 RAM

在哈佛架构的器件中，数据和程序存储器使用不同的总线，因而可同时访问这两种存储空间。

第 6.0 节“闪存程序存储器”提供了关于闪存程序存储器操作的更多详细信息。

5.1 程序存储器构成

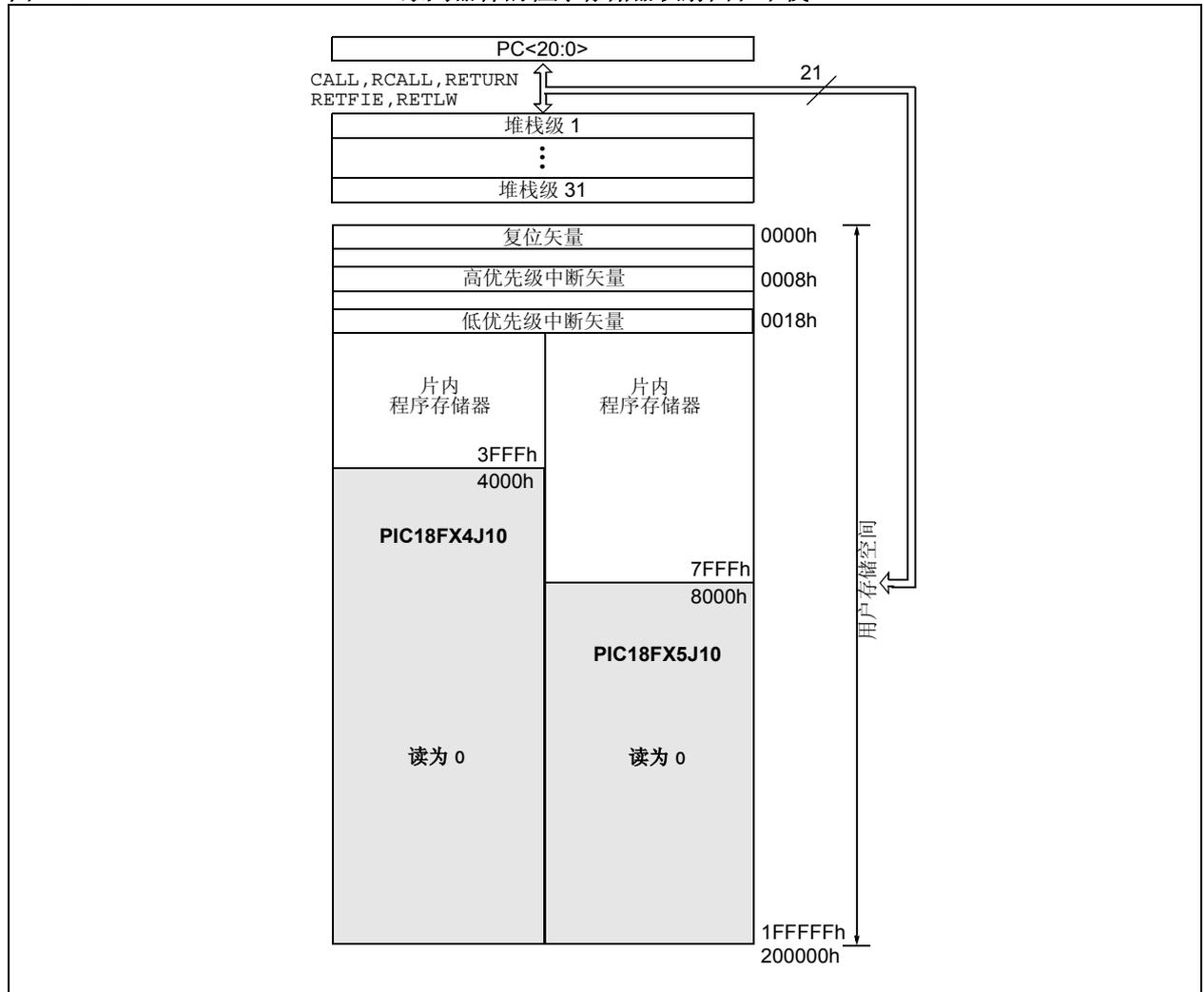
PIC18 单片机具有一个 21 位程序计数器，可以对 2MB 的程序存储空间进行寻址。访问物理实现的存储器的上边界和 2MB 地址之间的单元将会返回全 0（相当于执行 NOP 指令）。

PIC18F24J10 和 PIC18F44J10 具有 16 KB 的闪存存储器，能够存储多达 8,192 条单字指令。PIC18F25J10 和 PIC18F45J10 具有 32 KB 的闪存存储器，能够存储多达 16,384 条单字指令。

PIC18 器件有两个中断矢量。复位矢量地址为 0000h，中断矢量地址为 0008h 和 0018h。

图 5-1 给出了 PIC18F45J10 系列器件的程序存储器映射图。

图 5-1: PIC18F45J10 系列器件的程序存储器映射图和堆栈



PIC18F45J10 系列

5.1.1 程序计数器

程序计数器 (Program Counter, PC) 指定要取出执行的指令地址。PC 为 21 位宽, 被存放在 3 个独立的 8 位寄存器中。其中的低字节称为 PCL 寄存器, 该寄存器可读写。高字节, 即 PCH 寄存器, 包含 PC<15:8> 位, 不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。最高字节称为 PCU。该寄存器包含 PC<20:16> 位, 它也不能直接读写。通过 PCLATU 寄存器更新 PCU 寄存器。

通过对 PCL 执行写操作, 可以将 PCLATH 和 PCLATU 的内容传送到程序计数器。类似的, 通过对 PCH 执行读操作, 可以将程序计数器的两个高字节传送到 PCLATH 和 PCLATU。这对于计算 PC 的偏移量很有用处 (见第 5.1.4.1 节 “计算 GOTO”)。

PC 在程序存储器中按字节寻址。为防止 PC 不能正确获取字指令, 需要将 PCL 的最低有效位固定为 0。PC 每次加 2 来连续寻址程序存储器中的指令。

CALL、RCALL、GOTO 和程序转移指令将指令地址直接写入程序计数器。对于这些指令, PCLATH 和 PCLATU 的内容将不会被传送到程序计数器。

5.1.2 返回地址堆栈

用于存放返回地址的堆栈允许保存最多 31 个程序调用地址和中断矢量。当执行 CALL 或 RCALL 指令, 或者响应中断时, PC 值被压入堆栈。在执行 RETURN、RETLW 或 RETFIE 指令时 PC 值从堆栈弹出。PCLATU 和 PCLATH 不受 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和 5 位的堆栈指针 (STKPTR) 来实现 31 级的堆栈操作。堆栈既不占用程序存储空间也不占用数据存储空间。堆栈指针可以读写, 并且通过栈顶的特殊文件寄存器可以读写栈顶地址。也可使用这些寄存器将数据压入堆栈, 或将数据从堆栈弹出。

执行 CALL 类型的指令引起进栈操作: 堆栈指针首先加 1, 并且将 PC 的内容写入堆栈指针指向的单元 (PC 已经指向 CALL 后的指令)。执行 RETURN 类型的指令时, 引起出栈操作: STKPTR 寄存器所指向的单元的内容被传送给 PC, 然后堆栈指针减 1。

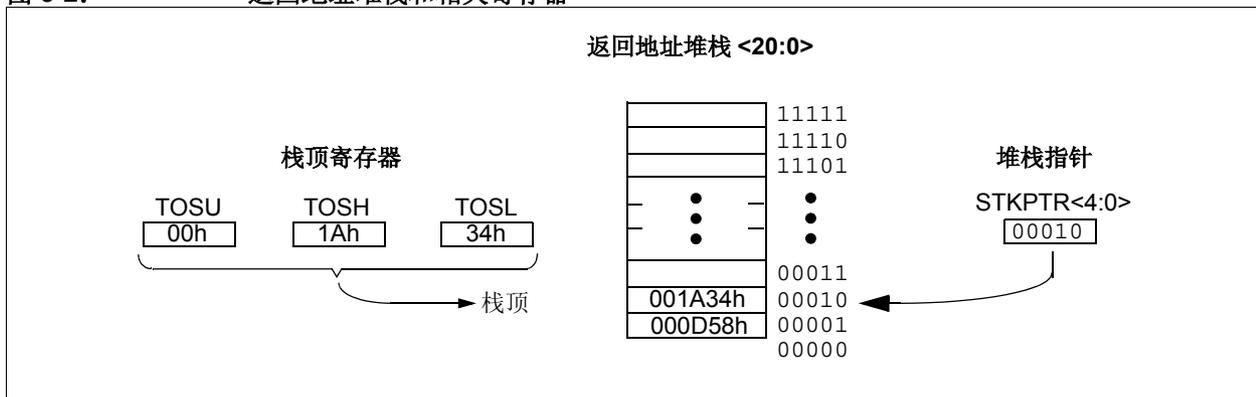
所有复位后, 堆栈指针均被初始化为 00000。堆栈指针值 00000 不指向任何 RAM 单元, 它只是一个复位值。状态位表明堆栈是满、上溢还是下溢。

5.1.2.1 栈顶访问

只有栈顶 (Top-of-Stack, TOS) 是可读写的。有 3 个寄存器 TOSU:TOSH:TOSL 用于保存 STKPTR 寄存器 (图 5-2) 所指向的堆栈单元的内容。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或中断后, 软件可以通过读取 TOSU:TOSH:TOSL 寄存器来读取进栈值。这些值可以被置入用户定义的软件堆栈。返回时, 软件将这些值存回 TOSU:TOSH:TOSL 并执行返回。

为防止对堆栈的意外操作, 访问堆栈时用户必须禁止全局中断使能位。

图 5-2: 返回地址堆栈和相关寄存器



5.1.2.2 返回堆栈指针 (STKPTR)

STKPTR 寄存器 (寄存器 5-1) 包含堆栈指针值、STKFUL (堆栈满) 状态位和 STKUNF (堆栈下溢) 状态位。堆栈指针值可为 0 到 31 之间的整数。向堆栈压入值前, 堆栈指针加 1; 而从堆栈弹出值后, 堆栈指针减 1。复位时, 堆栈指针值为零。用户可以读写堆栈指针的值。实时操作系统 (Real-Time Operating System, RTOS) 可以利用此特性对返回堆栈进行维护。

当向堆栈压入 PC 值 31 次 (且没有值从堆栈弹出) 后, STKFUL 位就会置 1。通过软件或 POR 使 STKFUL 位清零。

堆栈满时执行的操作由 STVREN (堆栈上溢复位使能) 配置位的状态决定。(有关器件配置位的介绍, 请参见第 20.1 节“配置位”)。如果 STVREN 位已经置 1 (默认), 第 31 次进栈将把 (PC+2) 值压入堆栈, 将 STKFUL 位置 1, 并复位器件。STKFUL 位将保持置 1, 而堆栈指针将被清零。

如果 STVREN 位被清零, 第 31 次进栈时 STKFUL 位会被置 1, 堆栈指针则加 1 变为 31。任何其他进栈操作都不会覆盖第 31 次进栈的值, 并且 STKPTR 将保持为 31。

当堆栈弹出次数足够卸空堆栈时, 下一次出栈会向 PC 返回一个零值, 并将 STKUNF 位置 1, 而堆栈指针则保持为 0。STKUNF 位将保持置 1, 直到被软件清零或发生 POR。

注: 下溢会导致向 PC 返回一个零值, 并使程序指向复位矢量, 此时可以验证堆栈状态并采取相应的操作。这与复位不同, 因为 SFR 的内容不受影响。

5.1.2.3 PUSH 和 POP 指令

由于栈顶是可以读写的, 因此将值压入堆栈或从堆栈弹出值而不影响程序的正常执行是非常理想的。PIC18 指令集包括两条指令 PUSH 和 POP, 它们允许在软件控制下对 TOS 进行操作。可以通过修改 TOSU、TOSH 和 TOSL, 将数据或返回地址压入堆栈。

PUSH 指令将当前的 PC 值压入堆栈。先将堆栈指针加 1, 再将当前 PC 值装入堆栈。

POP 指令通过将堆栈指针减 1 来丢弃当前的 TOS 值。然后前一个进栈值成为 TOS 值。

寄存器 5-1:

STKPTR: 堆栈指针寄存器

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
STKFUL ⁽¹⁾	STKUNF ⁽¹⁾	—	SP4	SP3	SP2	SP1	SP0	
bit 7								bit 0

bit 7 **STKFUL:** 堆栈满标志位⁽¹⁾

1 = 堆栈满或溢出
0 = 堆栈未满或未溢出

bit 6 **STKUNF:** 堆栈下溢标志位⁽¹⁾

1 = 发生堆栈下溢
0 = 未发生堆栈下溢

bit 5 **未用:** 读为 0

bit 4-0 **SP4:SP0:** 堆栈指针地址位

注 1: 通过用户软件或 POR 清零 bit 7 和 bit 6。

图注:

R = 可读位	W = 可写位	U = 未用位	C = 只可清零位
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

PIC18F45J10 系列

5.1.2.4 堆栈满和下溢复位

通过将配置寄存器 4L 中的 STVREN 位置 1，可以在出现堆栈溢出和堆栈下溢条件时使能器件复位。当 STVREN 位置 1 时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，然后使器件复位。当 STVREN 位清零时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，但不会使器件复位。通过用户软件或上电复位使 STKFUL 或 STKUNF 位清零。

5.1.3 快速寄存器堆栈

为 STATUS、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断矢量处执行指令时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE 和 FAST 指令从中断返回，这些寄存器中的值会被重新装入对应的寄存器中。

如果同时使能了低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储在堆栈寄存器中的值将被覆盖。在这种情况下，用户必须在低优先级中断期间用软件保存关键寄存器的值。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复 STATUS、WREG 和 BSR 寄存器。要将快速寄存器堆栈用于子程序调用，必须执行 CALL label 和 FAST 指令将 STATUS、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。然后执行 RETURN 和 FAST 指令，从快速寄存器堆栈中弹出并恢复这些寄存器的值。

例 5-1 给出了一个在子程序调用和返回期间使用快速寄存器堆栈的源代码示例。

例 5-1: 快速寄存器堆栈代码示例

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                    ;SAVED IN FAST REGISTER
                    ;STACK
    .
    .
SUB1    .
    .
        RETURN, FAST ;RESTORE VALUES SAVED
                    ;IN FAST REGISTER STACK
```

5.1.4 程序存储器中的查找表

有些编程需要在程序存储器中创建数据结构或查找表。对于 PIC18 器件，有两种方法可以实现查找表：

- 计算 GOTO
- 表读

5.1.4.1 计算 GOTO

计算 GOTO 是通过向程序计数器加一个偏移量来实现的。例 5-2 给出了一个示例。

使用 ADDWF PCL 指令和一组 RETLW nn 指令可以创建一个查找表。在调用该表前，会先将查找表的偏移量装入 W 寄存器。被调用程序的第一条指令是 ADDWF PCL 指令。接下去执行的是一条 RETLW nn 指令，它将数值 nn 返回给调用函数。

偏移量（WREG 中的值）指定程序计数器应该增加的字节数，其值应当为 2 的倍数（LSb = 0）。

在这种方法中，每个指令单元只能存储一个数据字节，并且要求返回地址堆栈还有空闲单元。

例 5-2: 使用偏移量的计算 GOTO

```
MOVWF  OFFSET, W
CALL    TABLE
ORG     nn00h
TABLE  ADDWF  PCL
        RETLW nnh
        RETLW nnh
        RETLW nnh
        .
        .
        .
```

5.1.4.2 表读和表写

有一种更好的方法可以将数据存储在程序存储器中，该方法允许在每个指令单元存储 2 个字节的数据。

使用表读和表写，每个程序字可以存储 2 个字节的查找表数据。表指针寄存器（Table Pointer, TBLPTR）指定字节地址，而表锁存器（Table Latch, TABLAT）储存从程序存储器读取或写入的数据。写入 / 读出程序存储器的数据每次为一个字节。

第 6.1 节“表读和表写”中将进一步讨论表读和表写操作。

5.2 PIC18 指令周期

5.2.1 时钟分配

单片机时钟输入信号，无论来自内部或外部时钟源，都会在器件内部被 4 分频用来产生 4 个不重叠的正交时钟信号（Q1、Q2、Q3 和 Q4）。程序计数器在每个 Q1 递增，并在 Q4 期间从程序存储器取指并将指令锁存到指令寄存器中。指令的译码和执行在下一个 Q1 到 Q4 周期完成。图 5-3 所示为时钟和指令执行的流程图。

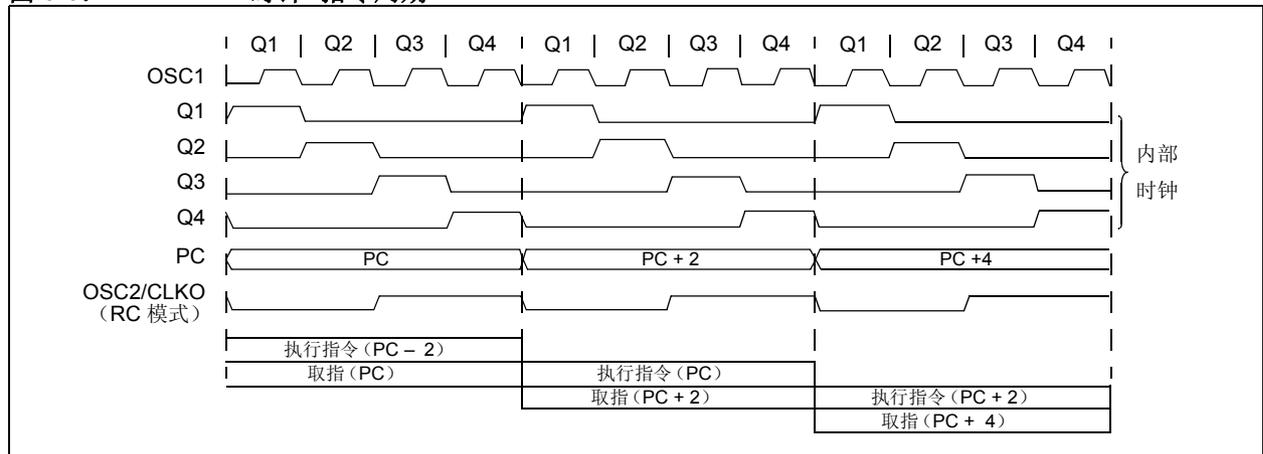
5.2.2 指令流 / 流水线

一个“指令周期”由 4 个 Q 周期组成，即 Q1 到 Q4。指令的取指和执行是以流水线的形式进行的，用一个指令周期来取指，而用另一个指令周期译码和执行指令。但由于是流水线操作，因此每条指令的等效执行时间都是一个指令周期。如果某条指令改变了程序计数器的值（如 GOTO 指令），则需要两个指令周期才能完成该指令（见例 5-3）。

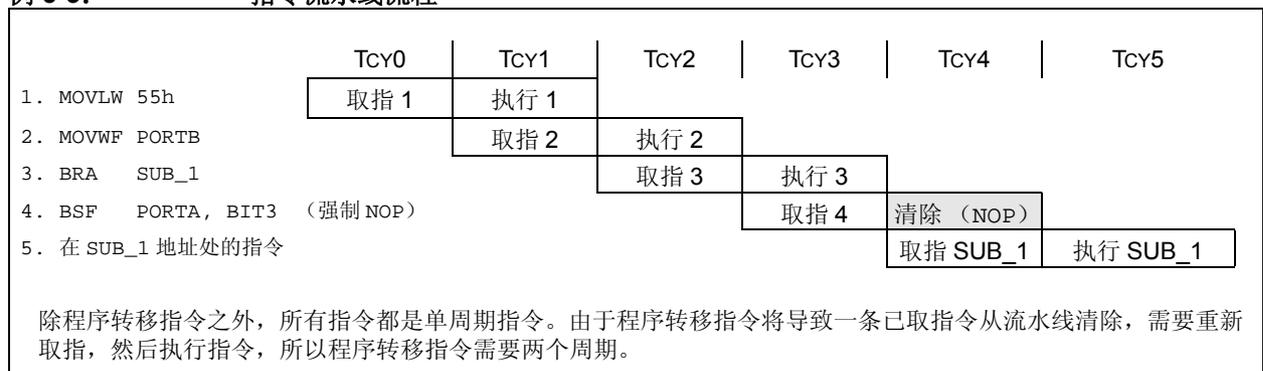
在 Q1 周期，程序计数器（PC）递增，开始取指。

指令的执行过程：在 Q1 周期，将所取指令锁存到指令寄存器（Instruction Register, IR）。然后在 Q2、Q3 和 Q4 周期中进行指令的译码和执行。其中读数据存储器（读操作数）发生在 Q2 周期，写操作发生在 Q4 周期（写目标地址）。

图 5-3: 时钟 / 指令周期



例 5-3: 指令流水线流程



PIC18F45J10 系列

5.2.3 程序存储器中的指令

程序存储器按字节寻址。指令以 2 字节或 4 字节形式存储在程序存储器中。指令字的最低有效字节始终存储在地址为偶数的程序存储器单元中 (LSB = 0)。要保证正确指向指令单元, PC 必须以 2 为单位递增, 并且 LSB 总是 0 (见第 5.1.1 节“程序计数器”)。

图 5-4 给出了指令字如何存储在程序存储器中的示例。

CALL 和 GOTO 指令在指令中嵌入了程序存储器的绝对地址。指令总是存储为一个字长, 因而指令所包含的数据为字地址。字地址会写入 PC<20:1>, 由 PC 在程序存储器中访问目标地址。图 5-4 中的指令 2 说明了指令 GOTO 0006h 在程序存储器中的译码过程。程序转移指令也采取同样的方式对相对地址偏移量进行译码。在转移指令中的偏移量代表单字指令数, PC 将以此作为偏移量跳转到指定的地址单元。第 21.0 节“指令集综述”提供了指令集的更多详情。

图 5-4: 程序存储器中的指令

程序存储器 字节单元 →			LSB = 1	LSB = 0	字地址 ↓
				000002h	
				000004h	
				000006h	
指令 1:	MOVLW	055h	0Fh	55h	000008h
指令 2:	GOTO	0006h	EFh	03h	00000Ah
			F0h	00h	00000Ch
指令 3:	MOVFF	123h, 456h	C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

5.2.4 双字指令

标准的 PIC18 指令集有 4 条双字指令: CALL、MOVFF、GOTO 和 LSFR。在所有情况下, 这些指令第二个字的高 4 位总是 1111, 而其余 12 位是立即数数据, 通常为一个数据存储地址。

指令的高 4 位 1111 用于指定一个特殊的 NOP 指令。指令的正确执行顺序为: 执行完第一个字之后立即按顺序访问并使用第二个字中的数据。如果由于某些原因跳过

了第一个字并自行执行指令的第二个字, 则其效果将相当于执行了一条 NOP 指令。如果双字指令跟在更改 PC 的条件指令后, 就有必要执行此操作。例 5-4 说明了其执行过程。

注: 请参见第 5.6 节“PIC18 指令执行和扩展指令集”了解扩展指令集中双字指令的信息。

例 5-4: 双字指令

情形 1:		
目标代码	源代码	
0110 0110 0000 0000	TSTFSZ	REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2 ; No, skip this word
1111 0100 0101 0110		; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3 ; continue code
情形 2:		
目标代码	源代码	
0110 0110 0000 0000	TSTFSZ	REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2 ; Yes, execute this word
1111 0100 0101 0110		; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3 ; continue code

5.3 数据存储器的构成

注： 当使能 PIC18 扩展指令集时，数据存储器某些方面的操作会改变。如需更多信息，请参见第 5.5 节“数据存储器 and 扩展指令集”。

PIC18 器件中的数据存储器是以静态 RAM 的方式实现的。在数据存储器中，每个寄存器有 12 位地址，数据存储容量可达 4096 个字节。存储空间被分为 16 个存储区，每个存储区包含 256 个字节；PIC18F45J10 系列器件使用全部 16 个存储区。图 5-5 显示了 PIC18F45J10 系列器件的数据存储器构成。

数据存储器由特殊功能寄存器（SFR）和通用寄存器（General Purpose Register, GPR）组成。SFR 用于单片机和外设功能模块的控制和状态显示，而 GPR 则用于在用户应用程序中存储数据和高速暂存操作。任何未用单元的读取值均为 0。

此指令集和架构支持跨存储区的操作。可以通过直接、间接或变址寻址模式访问整个数据存储器。本章后面的部分将讨论寻址模式。

为了确保能在一个周期内存取常用寄存器（SFR 和所选的 GPR），PIC18 器件设置了快速操作存储区（Access Bank）。这是一个 256 字节的存储空间，它可实现对 SFR 和 GPR 存储区 0 的部分低地址单元的快速存取，而无需使用 BSR。第 5.3.2 节“快速操作存储区”提供了对快速操作 RAM 的详细说明。

5.3.1 存储区选择寄存器（BSR）

存储容量较大的数据存储器需要有效的寻址机制，以便对所有地址进行快速存取。理想状况下，这意味着不必为每次读写操作提供完整地址。PIC18 器件是使用 RAM 区存储机制实现快速存取的。该机制将存储器空间分成连续的 16 个 256 字节的存储区。根据不同的指令，可以通过完整的 12 位地址直接寻址每个单元，或通过 8 位的低字节地址和 4 位存储区指针间接寻址每个单元。

PIC18 指令集中的大部分指令都使用存储区指针，也就是存储区选择寄存器（Bank Select Register, BSR）。SFR 保存单元地址的高 4 位，而指令本身则包括单元地址的低 8 位。只使用 BSR 的低 4 位（BSR3:BSR0），而不使用高 4 位，它们总是读为 0 且不能被写入。可以通过使用 MOVLB 指令直接装载 BSR。

BSR 的值指定数据存储器中的存储区。指令中的 8 位指向该存储区中的存储单元，可以将它看作是存储区下边界为起点的偏移量。图 5-6 所示是 BSR 的值与存储区之间的关系。

由于最多有 16 个寄存器共享同一个低位地址，用户必须非常小心以确保在执行数据读或写之前选择了正确的存储区。例如，当 BSR 为 0Fh 时，将程序数据写入地址为 F9h 的存储单元将中止程序的运行，并使程序计数器复位。

当选择存储区时，只有实际可使用的存储区才可以被读写。对不可用的存储区的写入将被忽略，而读不可用的存储区会返回 0。虽然是这样，STATUS 寄存器仍然会受到影响。图 5-5 中的数据存储器映射图指出了可使用的存储区。

在 PIC18 的内核指令集中，只有 MOVFF 指令指定源寄存器和目标寄存器的完整 12 位地址。此指令在执行时完全忽略 BSR。所有其他指令仅包含作为操作数的低位地址，而且必须使用 BSR 或快速操作存储区来寻址目标寄存器。

PIC18F45J10 系列

图 5-5: PIC18F24J10/44J10 器件的数据存储器映射图

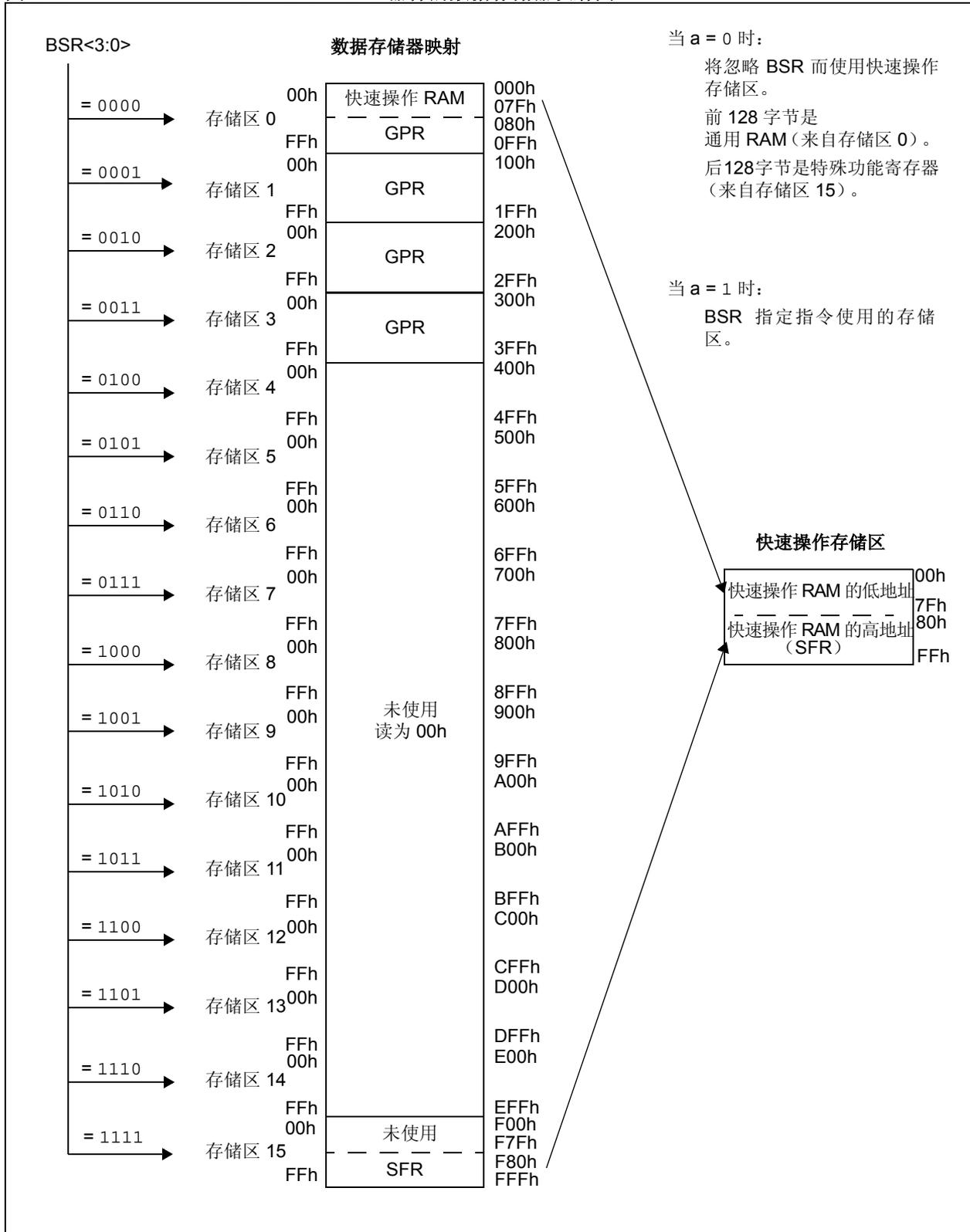
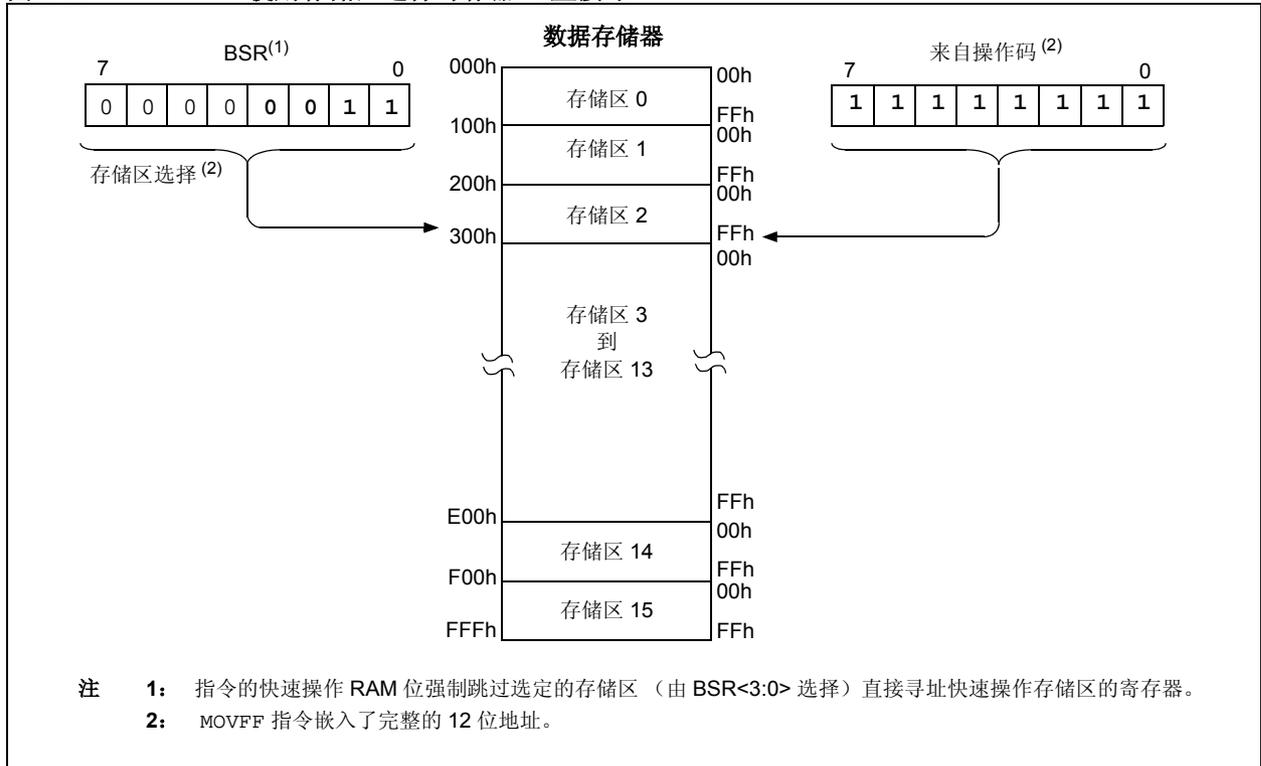


图 5-6: 使用存储区选择寄存器 (直接寻址)



5.3.2 快速操作存储区

使用 BSR 和指令内嵌的 8 位地址可以使用户对整个数据存储空间进行寻址，这同时意味着用户必须始终确保了正确的存储区。否则，可能从错误的单元读取数据或将数据写入错误的单元。如果本来是向 GPR 进行写操作却将结果写入到了 SFR，后果是非常严重的。但是在每次向数据存储器进行读或写操作时，验证和 / 或更改 BSR 可能会降低代码的执行效率。

为了连续访问大多数常用的数据存储单元，必须为数据存储器配置快速操作存储区，这使得用户无需指定 BSR 即可访问被映射的存储区。快速操作存储区由存储区 0 的前 128 个字节 (00h-7Fh) 和存储区 15 的后 128 个字节 (80h-FFh) 组成。低半部分就是“快速操作 RAM”，由 GPR 组成。地址较高的部分则被映射为器件的 SFR。这两个区域可以在快速操作存储区中连续映射并且可以用 8 位地址进行线性寻址 (图 5-5)。

快速操作存储区供包括快速操作 RAM 位 (指令中的“a”参数) 的 PIC18 内核指令使用。当“a”等于 1 时，指令使用 BSR 和包含在操作码中的 8 位地址来对数据存储器进行寻址。但是当“a”为 0 时，指令被强制使用快速操作存储区地址映射；BSR 的当前值被忽略。

该“强制”寻址方式可使指令在一个周期内对数据地址进行操作，而无需首先更新 BSR。这意味着用户可以更有效地对 8 位地址为 80h 及以上的 SFR 进行取值和操作。地址低于 80h 的快速操作 RAM 非常适合于存储那些用户可能需要快速存取的数据 (如直接计算结果或常用程序变量)。快速操作 RAM 还可以实现更快速、代码效率更高的现场保护和变量切换。

当使能扩展的指令集 (XINST 配置位 = 1) 时，快速操作存储区的映射略有不同。第 5.5.3 节“在立即数变址寻址模式下映射快速操作存储区”更详细地讨论了此操作。

5.3.3 通用寄存器

PIC18 器件可能在 GRP 区中划分了一部分存储区。这部分存储区为数据 RAM，所有指令都可以访问它。GPR 从存储区 0 的底部 (地址 000h) 开始，向上直到 SFR 区的底部。上电复位不会将 GPR 初始化，并且其他复位也不会改变其内容。

PIC18F45J10 系列

5.3.4 特殊功能寄存器

特殊功能寄存器（Special Function Registers, SFR）是 CPU 和外设模块用来控制器件操作的寄存器。这类寄存器以静态 RAM 的形式实现。SFR 起始于数据存储器的顶部（FFFh）并且向下扩展到存储区 15 的上半部分（F80h 到 FFFh）。表 5-1 和表 5-2 列出了这些寄存器。

可以将 SFR 归为两类：与“内核”器件功能（ALU、复位和中断）相关的寄存器和与外设功能相关的寄存器。复位和中断寄存器在它们各自的章节中说明，而 ALU 的 STATUS 寄存器在本章稍后说明。与外设功能相关的寄存器将在该外设的章节中进行说明。

SFR 通常位于被其控制的外设中。未使用的 SFR 单元不可用，它们的读取值为 0。

表 5-1: PIC18F45J10 系列器件的特殊功能寄存器映射图

地址	名称	地址	名称	地址	名称	地址	名称
FFFh	TOSU	FDfH	INDF2 ⁽¹⁾	FBfH	CCPR1H	F9fH	IPR1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCCh	CCPR2H	F9Ch	— ⁽²⁾
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	— ⁽²⁾
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	— ⁽²⁾
FF9h	PCL	FD9h	FSR2L	FB9h	— ⁽²⁾	F99h	— ⁽²⁾
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	— ⁽²⁾
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL ⁽³⁾	F97h	— ⁽²⁾
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS ⁽³⁾	F96h	TRISE ⁽³⁾
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD ⁽³⁾
FF4h	PRODH	FD4h	— ⁽²⁾	FB4h	CMCON	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	— ⁽²⁾	F93h	TRISB
FF2h	INTCON	FD2h	— ⁽²⁾	FB2h	— ⁽²⁾	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	— ⁽²⁾	F91h	— ⁽²⁾
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	— ⁽²⁾
FEFh	INDF0 ⁽¹⁾	FCfH	TMR1H	FAfH	SPBRG	F8fH	— ⁽²⁾
FEeh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAeh	RCREG	F8eh	SSP2BUF
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8dh	LATE ⁽³⁾
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACCh	TXSTA	F8Ch	LATD ⁽³⁾
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	— ⁽²⁾	F8Ah	LATB
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	— ⁽²⁾	F89h	LATA
FE8h	WREG	FC8h	SSP1ADD	FA8h	— ⁽²⁾	F88h	SSP2ADD ⁽³⁾
FE7h	INDF1 ⁽¹⁾	FC7h	SSP1STAT	FA7h	EECON2 ⁽¹⁾	F87h	SSP2STAT ⁽³⁾
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSP1CON1	FA6h	EECON1	F86h	SSP2CON1 ⁽³⁾
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSP1CON2	FA5h	IPR3	F85h	SSP2CON2 ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA

- 注
- 1: 这不是物理寄存器。
 - 2: 不存在的寄存器，读取值为 0。
 - 3: 28 引脚器件上没有这些寄存器。

表 5-2: PIC18F24J10/25J10/44J10/45J10 寄存器汇总

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	上电复位与 欠压复位时的 值	详情请见: (页)
TOSU	—	—	—	栈顶最高字节 (TOS<20:16>)					---0 0000	43, 48
TOSH	栈顶高字节 (TOS<15:8>)								0000 0000	43, 48
TOSL	栈顶低字节 (TOS<7:0>)								0000 0000	43, 48
STKPTR	STKFUL	STKUNF	—	返回堆栈指针					00-0 0000	43, 49
PCLATU	—	—	—	PC<20:16> 的保持寄存器					---0 0000	43, 48
PCLATH	PC<15:8> 的保持寄存器								0000 0000	43, 48
PCL	PC 低字节 (PC<7:0>)								0000 0000	43, 48
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 (TBLPTR<20:16>)					--00 0000	43, 70
TBLPTRH	程序存储器表指针高字节 (TBLPTR<15:8>)								0000 0000	43, 70
TBLPTRL	程序存储器表指针低字节 (TBLPTR<7:0>)								0000 0000	43, 70
TABLAT	程序存储器表锁存器								0000 0000	43, 70
PRODH	乘积寄存器高字节								XXXX XXXX	43, 77
PRODL	乘积寄存器低字节								XXXX XXXX	43, 77
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	43, 81
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1	43, 82
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00	43, 83
INDF0	使用 FSR0 的内容来寻址数据存储器—FSR0 的值不变 (不是物理寄存器)								N/A	43, 62
POSTINC0	使用 FSR0 的内容来寻址数据存储器—FSR0 的值后增 (不是物理寄存器)								N/A	43, 62
POSTDEC0	使用 FSR0 的内容来寻址数据存储器—FSR0 的值后减 (不是物理寄存器)								N/A	43, 62
PREINC0	使用 FSR0 的内容来寻址数据存储器—FSR0 的值预增 (不是物理寄存器)								N/A	43, 62
PLUSW0	使用 FSR0 的内容来寻址数据存储器—FSR0 的值预增 (不是物理寄存器), FSR0 的偏移量由 W 寄存器提供								N/A	43, 62
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高字节				---- XXXX	43, 62
FSR0L	间接数据存储器地址指针 0 的低字节								XXXX XXXX	43, 62
WREG	工作寄存器								XXXX XXXX	43
INDF1	使用 FSR1 的内容来寻址数据存储器—FSR1 的值不变 (不是物理寄存器)								N/A	43, 62
POSTINC1	使用 FSR1 的内容来寻址数据存储器—FSR1 的值后增 (不是物理寄存器)								N/A	43, 62
POSTDEC1	使用 FSR1 的内容来寻址数据存储器—FSR1 的值后减 (不是物理寄存器)								N/A	43, 62
PREINC1	使用 FSR1 的内容来寻址数据存储器—FSR1 的值预增 (不是物理寄存器)								N/A	43, 62
PLUSW1	使用 FSR1 的内容来寻址数据存储器—FSR1 的值预增 (不是物理寄存器), FSR1 的偏移量由 W 寄存器提供								N/A	43, 62
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高字节				---- XXXX	43, 62
FSR1L	间接数据存储器地址指针 1 的低字节								XXXX XXXX	43, 62
BSR	—	—	—	—	存储区选择寄存器				---- 0000	43, 53
INDF2	使用 FSR2 的内容来寻址数据存储器—FSR2 的值不改变 (不是物理寄存器)								N/A	44, 62
POSTINC2	使用 FSR2 的内容来寻址数据存储器—FSR2 的值后增 (不是物理寄存器)								N/A	44, 62
POSTDEC2	使用 FSR2 的内容来寻址数据存储器—FSR2 的值后减 (不是物理寄存器)								N/A	44, 62
PREINC2	使用 FSR2 的内容来寻址数据存储器—FSR2 的值预增 (不是物理寄存器)								N/A	44, 62
PLUSW2	使用 FSR2 的内容来寻址数据存储器—FSR2 的值预增 (不是物理寄存器), FSR2 的偏移量由 W 寄存器提供								N/A	44, 62
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高字节				---- XXXX	44, 62
FSR2L	间接数据存储器地址指针 2 的低字节								XXXX XXXX	44, 62
STATUS	—	—	—	N	OV	Z	DC	C	---x XXXX	44, 60

图注: x = 未知, u = 不变, — = 未用位, α = 取值视具体情况而定
注 1: 请参见第 4.4 节“欠压复位 (BOR) (仅 PIC18F2XJ10/4XJ10 器件)”。
注 2: 这些寄存器和 / 或在 28 引脚器件上不存在并且读为 0。此处给出了 40/44 引脚器件的复位值。未用位以“—”表示。

PIC18F45J10 系列

表 5-2: PIC18F24J10/25J10/44J10/45J10 寄存器汇总 (续)

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	上电复位与 欠压复位时 的值	详情请见: (页)
TMR0H	Timer0 寄存器的高字节								0000 0000	44, 113
TMR0L	Timer0 寄存器的低字节								xxxx xxxx	44, 113
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	44, 111
OSCCON	IDLEN	—	—	—	OSTS	—	SCS1	SCS0	0--- q-00	28, 44
WDTCON	—	—	—	—	—	—	—	SWDTEN	--- --0	44, 235
RCON	IPEN	—	—	R̄I	T̄O	PD	POR	BOR ⁽¹⁾	0--1 11q0	38, 42, 90
TMR1H	Timer1 寄存器的高字节								xxxx xxxx	44, 119
TMR1L	Timer1 寄存器的低字节								xxxx xxxx	44, 119
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN̄C	TMR1CS	TMR1ON	0000 0000	44, 115
TMR2	Timer2 寄存器								0000 0000	44, 122
PR2	Timer2 周期寄存器								1111 1111	44, 122
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-00 0000	44, 121
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								xxxx xxxx	44, 154
SSP1ADD	MSSP1 地址寄存器 (I ² C™ 从动模式)。MSSP1 波特率重载寄存器 (I ² C 主控模式)								0000 0000	44, 155
SSP1STAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	0000 0000	44, 146, 156
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	44, 147, 157
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	44, 158
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	44, 218
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	44, 218
ADCON0	ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0-00 0000	44, 209
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0qqq	44, 210
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	44, 211
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								xxxx xxxx	45, 124
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								xxxx xxxx	45, 124
CCP1CON	P1M1 ⁽²⁾	P1M0 ⁽²⁾	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	45, 123,
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								xxxx xxxx	45, 124
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								xxxx xxxx	45, 124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	45, 123
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	45, 190
ECCP1DEL	PRSEN	PDC6 ⁽²⁾	PDC5 ⁽²⁾	PDC4 ⁽²⁾	PDC3 ⁽²⁾	PDC2 ⁽²⁾	PDC1 ⁽²⁾	PDC0 ⁽²⁾	0000 0000	45, 140
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 ⁽²⁾	PSSBD0 ⁽²⁾	0000 0000	45, 141
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	45, 225
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	45, 219

图注:

x = 未知, u = 不变, — = 未用位, q = 取值视具体情况而定

注 1: 请参见第 4.4 节“欠压复位 (BOR) (仅 PIC18F2XJ10/4XJ10 器件)”。

注 2: 这些寄存器和 / 或在 28 引脚器件上不存在并且读为 0。此处给出了 40/44 引脚器件的复位值。未用位以“—”表示。

PIC18F45J10 系列

表 5-2: PIC18F24J10/25J10/44J10/45J10 寄存器汇总 (续)

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	上电复位与 欠压复位时 的值	详情请见: (页)
SPBRGH	EUSART 波特率发生器寄存器的高字节								0000 0000	45, 192
SPBRG	EUSART 波特率发生器寄存器的低字节								0000 0000	45, 192
RCREG	EUSART 接收寄存器								0000 0000	45, 199
TXREG	EUSART 发送寄存器								xxxx xxxx	45, 197
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	45, 188
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	45, 189
EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								0000 0000	45, 68
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	---0 x00-	45, 69
IPR3	SSP2IP	BCL2IP	—	—	—	—	—	—	11-- ----	45, 89
PIR3	SSP2IF	BCL2IF	—	—	—	—	—	—	00-- ----	45, 85
PIE3	SSP2IE	BCL2IE	—	—	—	—	—	—	00-- ----	45, 87
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	11-- 1--1	45, 89
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	00-- 0--0	45, 85
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	00-- 0--0	45, 87
IPR1	PSP1P ⁽²⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	45, 88
PIR1	PSP1IF ⁽²⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	45, 84
PIE1	PSP1IE ⁽²⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	0000 0000	45, 86
TRISE ⁽²⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	1111 -111	46, 107
TRISD ⁽²⁾	PORTD 数据方向控制寄存器								1111 1111	46, 103
TRISC	PORTC 数据方向控制寄存器								1111 1111	46, 100
TRISB	PORTB 数据方向控制寄存器								1111 1111	46, 97
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	--1- 1111	46, 94
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								xxxx xxxx	46, 154
LATE ⁽²⁾	—	—	—	—	—	PORTE 数据锁存器 (读取和写入数据锁存器)		—	---- -xxx	46, 106
LATD ⁽²⁾	PORTD 数据锁存器 (读取和写入数据锁存器)								xxxx xxxx	46, 103
LATC	PORTC 数据锁存器 (读取和写入数据锁存器)								xxxx xxxx	46, 100
LATB	PORTB 数据锁存器 (读取和写入数据锁存器)								xxxx xxxx	46, 97
LATA	—	—	PORTA 数据锁存器 (读取和写入数据锁存器)					—	--xx xxxx	46, 94
SSP2ADD	MSSP2 地址寄存器 (I ² C™ 从动模式)。MSSP2 波特率重载寄存器 (I ² C 主控模式)。								0000 0000	46, 154
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	46, 146, 156
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	46, 147, 157
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	46, 158
PORTE ⁽²⁾	—	—	—	—	—	RE2 ⁽²⁾	RE1 ⁽²⁾	RE0 ⁽²⁾	---- -xxx	46, 106
PORTD ⁽²⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	46, 103
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	46, 100
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	46, 97
PORTA	—	—	RA5	—	RA3	RA2	RA1	RA0	--0- 0000	46, 94

图注: x = 未知, u = 不变, — = 未用位, q = 取值视具体情况而定

注 1: 请参见第 4.4 节“欠压复位 (BOR) (仅 PIC18F2XJ10/4XJ10 器件)”。

注 2: 这些寄存器和 / 或在 28 引脚器件上不存在并且读为 0。此处给出了 40/44 引脚器件的复位值。未用位以“—”表示。

PIC18F45J10 系列

5.3.5 STATUS 寄存器

如寄存器 5-2 所示，STATUS 寄存器包含 ALU 的算术运算状态。和其他 SFR 一样，它可以是任何指令的操作数。

如果一条影响 Z、DC、C、OV 或 N 位的指令以 STATUS 寄存器作为目标寄存器，指令执行的结果将不会被直接写入，而是根据执行的指令来更新 STATUS 寄存器。所以，当执行一条把 STATUS 寄存器作为目标寄存器的指令后，STATUS 寄存器的结果可能和预想的不一樣。例如，执行 CLRF STATUS 将把 Z 位置 1，而其他状态位保持不变（000u u1uu）。

因此，建议仅使用 BCF、BSF、SWAPF、MOVFF 和 MOVWF 指令来改变 STATUS 寄存器，因为这些指令不会影响该寄存器中的 Z、C、DC、OV 或 N 位。

欲知其他不会影响状态位的指令，请参见表 21-2 和表 21-3 中的指令集综述。

注： 在减法运算中，C 和 DC 分别作为借位（borrow）和辅助借位（digit borrow）。

寄存器 5-2:

STATUS 寄存器

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
—	—	—	N	OV	Z	DC	C	
bit 7								bit 0

bit 7-5 未用：读为 0

bit 4 **N**：负标志位
此位用于有符号的算术运算（二进制补码）。它表示结果是否为负（ALU MSB = 1）。
1 = 结果为负
0 = 结果为正

bit 3 **OV**：溢出位
此位用于有符号的算术运算（二进制补码）。表明溢出了 7 位二进制数的范围，溢出将导致符号位（结果的 bit 7）发生改变。
1 = 有符号的算术运算中发生溢出（本次运算）
0 = 没有发生溢出

bit 2 **Z**：全零标志位
1 = 算术运算或逻辑运算结果为零
0 = 算术运算或逻辑运算结果不为零

bit 1 **DC**：辅助进位 / 借位位
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：
1 = 结果的第 4 个低有效位发生了进位
0 = 结果的第 4 个低有效位未发生进位

注： 对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码实现的。对于移位指令（RRF 或 RLF），此位的值来自源寄存器的 bit 4 或 bit 3。

bit 0 **C**：进位 / 借位位
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：
1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

注： 对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码实现的。对于移位指令（RRF 或 RLF），此位的值来自源寄存器的最高位或最低位。

图注：

R = 可读位	W = 可写位	U = 未用位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

5.4 数据寻址模式

注： 当使能 PIC18 扩展指令集时，PIC18 内核指令集中的某些指令的执行会发生改变。更多信息，请参见第 5.5 节“数据存储器和扩展指令集”。

虽然只能用一种方法（即通过程序计数器）对程序存储器进行寻址，但是可以用多种方法来对数据存储空间进行寻址。对于大多数指令，寻址模式是固定的。其他指令最多可以使用 3 种模式，具体使用哪一种模式取决于使用哪些操作数以及是否使能了扩展指令集。

寻址模式有：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

当使能扩展指令集（XINST 配置位 = 1）时，还可以使用另一种寻址模式，即立即数变址寻址模式。第 5.5.1 节“使用立即数偏移量进行变址寻址”将更详细地讨论其操作。

5.4.1 固有和立即数寻址

很多 PIC18 控制指令根本不需要任何参数，执行这些指令要么对整个器件造成影响，要么仅针对一个寄存器进行操作。这种寻址模式就是固有寻址。例如 SLEEP、RESET 和 DAW 指令。

其他指令的工作方式与此类似但需要操作码中有直接的参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址。例如 ADDLW 和 MOVLW，它们分别将立即数值移入 W 寄存器或从中移出。其他的立即数寻址指令，例如 CALL 和 GOTO，包括一个 20 位的程序存储器地址。

5.4.2 直接寻址

直接寻址在操作码中指定操作的全部或部分源地址和 / 或目标地址。此选项由指令附带的参数指定。

在 PIC18 内核指令集中，面向位和字节的指令默认情况下使用直接寻址。所有这些指令都包含一个 8 位的直接地址作为它们的最低有效字节。此地址指定数据 RAM（第 5.3.3 节“通用寄存器”）的某个存储区中的寄存器地址或快速操作存储区（第 5.3.2 节“快速操作存储区”）中的单元作为指令的数据源。

快速操作 RAM 位“a”决定地址的解析方式。当“a”为 1 时，BSR（第 5.3.1 节“存储区选择寄存器（BSR）”）的内容和地址一起用于确定寄存器完整的 12 位地址。当“a”为 0 时，此地址将被解析为快速操作存储区中的一个寄存器。使用快速操作 RAM 寻址有时候也被称为直接强制寻址模式。

有几个指令，比如 MOVFF，在操作码中包含完整的 12 位地址（源或目标地址）。在这些情况下，完全忽略 BSR。

操作结果的目标寄存器由目标位“d”确定。当“d”为 1 时，结果被存储到源寄存器并覆盖它原来的内容。当“d”为 0 时，结果被存储在 W 寄存器中。不带“d”参数的指令的目标地址是隐含的，它们是操作的目标寄存器或 W 寄存器。

5.4.3 间接寻址

间接寻址允许用户访问数据存储区中的单元而不需要在指令中给出一个固定的地址。这是通过使用特殊文件寄存器（File Select Register, FSR）作为指针指向被读取或写入的单元实现的。由于 FSR 本身作为特殊功能寄存器位于 RAM 中，所以也可以在程序控制下直接对它们进行操作。这使得 FSR 对于在数据存储区中实现诸如表和数组等数据结构非常有用。

也可以使用间接文件操作数（Indirect File Operand, INDF）对寄存器进行间接寻址。这种操作允许自动递增、递减或偏移指针，从而自动控制指针的值。它通过循环提高代码执行效率，如例 5-5 给出的清零整个 RAM 存储区的例子。

例 5-5: 使用间接寻址清零 RAM（存储区 1）的方法

```

LFSR    FSR0, 100h ;
NEXT    CLRF    POSTINC0 ; Clear INDF
                                ; register then
                                ; inc pointer
        BTFSS   FSR0H, 1 ; All done with
                                ; Bank1?
        BRA     NEXT     ; NO, clear next
CONTINUE                                ; YES, continue
    
```

PIC18F45J10 系列

5.4.3.1 FSR 寄存器和 INDF 操作数

间接寻址的核心是三组寄存器：FSR0、FSR1和FSR2。每组寄存器都代表一对8位寄存器，FSRnH和FSRnL。FSRnH寄存器的高四位未使用，所以每对FSR只保存一个12位二进制数，从而可以线性寻址整个数据存储空间。因此，FSR寄存器对被用作数据存储器的地址指针。

间接寻址是通过一组间接文件操作数（从INDF0到INDF2）完成的。这些操作数可以被看作“虚拟”寄存器：它们是被映射到SFR空间中而不是通过物理方式实现的。对特定的INDF寄存器执行读或写操作实际上访问的是相应的FSR寄存器对。例如，读INDF1就是读FSR1H:FSR1L指向的地址单元中的数据。使用INDF寄存器作为操作数的指令实际上使用的是相应的FSR的内容，该内容为指向目标地址的指针。INDF操作数只是使用指针的一种较方便的方法。

由于间接寻址使用完整的12位地址，因此没有必要进行数据RAM分区。因此BSR的当前内容和快速操作RAM位对于确定目标地址没有影响。

5.4.3.2 FSR 寄存器和 POSTINC、POSTDEC、PREINC 以及 PLUSW

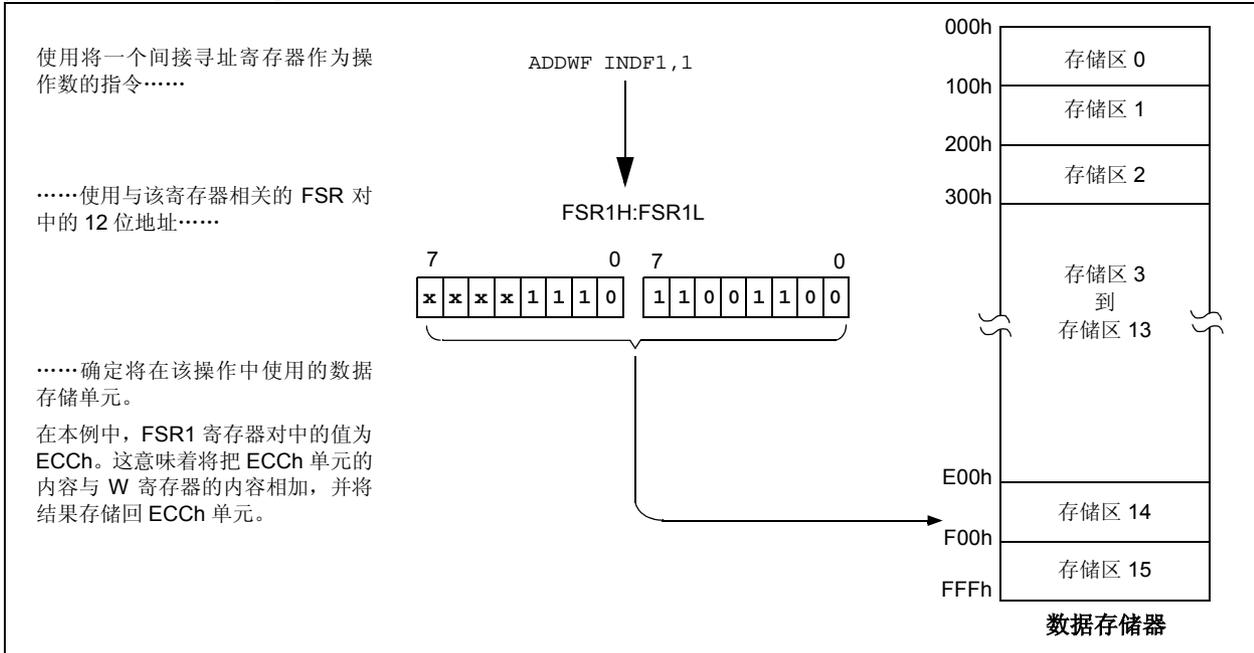
除了INDF操作数之外，每对FSR寄存器还有四个额外的间接操作数。和INDF一样，它们也是不能直接读写的“虚拟”寄存器。访问这些寄存器其实就是访问相关的FSR寄存器对，同时对其内容进行特定的操作。这些寄存器是：

- POSTDEC: 访问FSR值，然后自动将它减1
- POSTINC: 访问FSR值，然后自动将它加1
- PREINC: 将FSR的值加1，然后在操作中使用该值
- PLUSW: 将W寄存器中有符号的值（从-127到128）与FSR寄存器中的值相加，并在操作中使用得到的新值。

如前文所述，访问INDF寄存器实际使用的是FSR寄存器的值，但不会改变该值。同样，访问PLUSW寄存器是将W寄存器中的值作为FSR值的偏移量，该操作也不会改变这两个寄存器中的值。访问其他虚拟寄存器会更改FSR寄存器的值。

用POSTDEC、POSTINC和PREINC对FSR进行操作会影响整个寄存器，也就是说可能会发生FSRnL寄存器从FFh到00h溢出并向FSRnH寄存器进位。但这些操作的结果不会更改STATUS寄存器中的标志位（如Z、N和OV等）。

图 5-7: 间接寻址



PLUSW 寄存器可以用于在数据存储空间实现变址寻址。通过控制 W 寄存器中的值，用户可以访问相对当前指针地址有固定偏移量的地址单元。在某些应用中，该功能可以被用于在数据存储内部实现某些非常有用的程序控制结构，如软件堆栈。

5.4.3.3 通过 FSR 对其他 FSR 进行操作

在某些特殊情况下，间接寻址操作以其他 FSR 或虚拟寄存器作为寻址目标。例如，使用 FSR 指向一个虚拟寄存器会导致操作不成功。假设如下特殊情况：FSR0H:FSR0L 保存的是 INDF1 的地址 FE7h。尝试使用 INDF0 作为操作数读取 INDF1 的值，将返回 00h。尝试使用 INDF0 作为操作数写入 INDF1，将会导致执行一条 NOP 指令。

另一方面，使用虚拟寄存器对一对 FSR 寄存器进行写操作可能会产生与预期不同的结果。在这些情形下，会将值写入一对 FSR 寄存器，但 FSR 中的值不会有任何递增或递减。因此，写入 INDF2 或 POSTDEC2 时会把同样的值写入 FSR2H:FSR2L。

由于 FSR 是在 SFR 空间中映射的物理寄存器，所以可以通过直接寻址对它们进行操作。用户在使用这些寄存器时应特别小心，尤其是在代码使用间接寻址的情况下。

同样，通常允许通过间接寻址对所有其他的 SFR 进行操作。用户在进行此类操作时应特别小心，以免更改设置从而影响器件操作。

5.5 数据存储器和扩展指令集

使能 PIC18 扩展指令集 (XINST 配置位 = 1) 将显著更改数据存储及其寻址的方式。特别是许多 PIC18 内核指令使用快速操作存储区的方式会有所不同。这是由于引入了对数据存储空间的新的寻址模式造成的。

同样需要了解哪些部分保持不变。数据存储空间的大小及其线性寻址方式都不会改变。SFR 映射也保持不变。PIC18 内核指令也仍然以直接和间接寻址模式进行操作；固有和立即数指令操作依旧。FSR0 和 FSR1 的间接寻址方式也保持不变。

5.5.1 使用立即数偏移量进行变址寻址

使能 PIC18 扩展的指令集将更改使用 FSR2 寄存器对在快速操作 RAM 内进行间接寻址的方式。在适当的条件下，使用快速操作存储区的指令（即面向位和字节的指令）可以利用指令中的偏移量来执行变址寻址。这种特定的寻址模式被称为使用立即数偏移量的变址寻址或立即数变址寻址模式。

在使用扩展的指令集时，这种寻址模式有如下要求：

- 强制使用快速操作存储区 (“a” = 0)；且
- 指针地址参数要小于或等于 5Fh。

在这些条件下，指令的 f 地址不被解析为地址的低字节（在直接寻址中与 BSR 一起使用）或快速操作存储区中的 8 位地址。相反，该值被解析为由 FSR2 指定的地址指针的偏移量。该偏移量与 FSR2 的内容相加以获得操作的目标地址。

5.5.2 受立即数变址寻址模式影响的指令

任何使用直接寻址的 PIC18 内核指令均会受到立即数变址寻址模式的潜在影响，包括所有面向字节和位的指令，或标准 PIC18 指令集中几乎一半的指令。只能使用固有或立即数寻址模式的指令不受影响。

此外，如果面向字节和位的指令不使用快速操作存储区（快速操作 RAM 位为 “1”）或包含 60h 以上的地址，它们也不受影响。符合这些条件的指令会像以前一样执行。图 5-8 给出了当使能了扩展指令集时，各种寻址模式之间的对比。

那些想要在立即数变址寻址模式中使用面向字节或位指令的用户，应该注意此模式下汇编语法的改变。第 21.2.1 节“扩展指令的语法”中将对此进行更详细的说明。

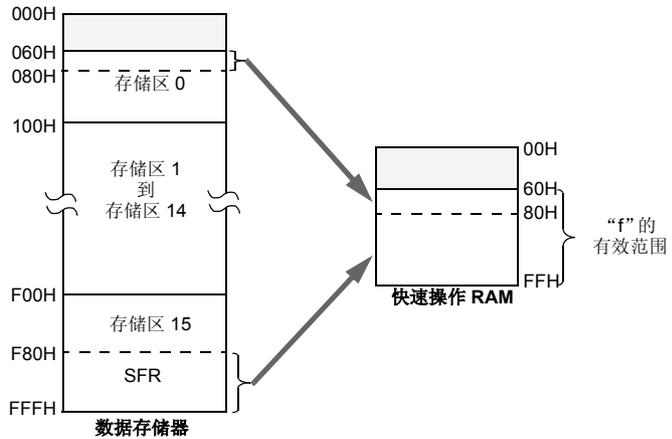
PIC18F45J10 系列

图 5-8: 面向位和字节的指令的寻址模式对比 (使能了扩展指令集)

指令示例: `ADDWF f, d, a` (操作码: `0010 01da ffff ffff`)

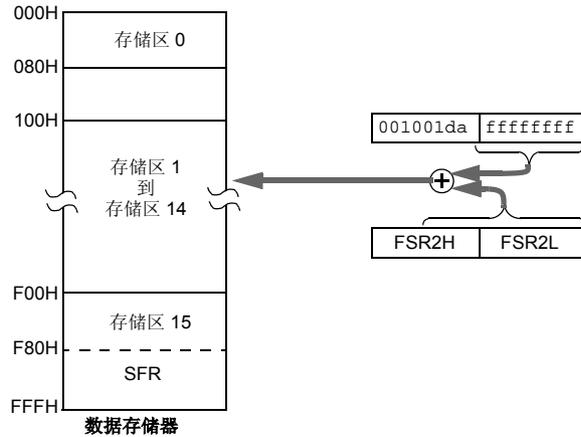
当 a = 0 且 f ≥ 60h:

此指令以直接强制模式执行。“f”被解析为快速操作 RAM 中 060h 和 0FFh 之间的单元地址。这实际上是从 060h 到 07Fh (存储区 0) 和从 F80h 到 FFFh (存储区 15) 的数据存储单元。



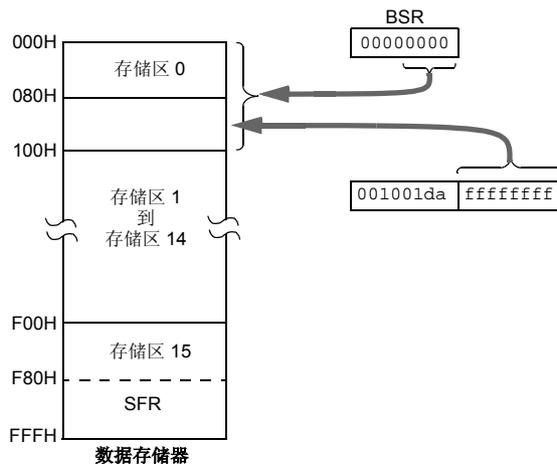
当 a = 0 且 f ≤ 5Fh:

该指令以立即数变址寻址模式执行。“f”被解析为 FSR2 中地址值的偏移量。这两个值相加以获得指令的目标寄存器的地址。此地址可以在数据存储空间的任何地方。注意在此模式中，正确的语法是：`ADDWF [k], d` 其中“k”就是“f”。



当 a = 1 时 (f 的所有值)

指令以直接模式执行 (也被称为直接长地址寻址模式)。“f”被解析为数据存储空间的 16 个存储区中的一个单元地址。存储区由存储区选择寄存器 (BSR) 指定。此地址可以在数据存储空间的任何位置。



5.5.3 在立即数变址寻址模式下映射快速操作存储区

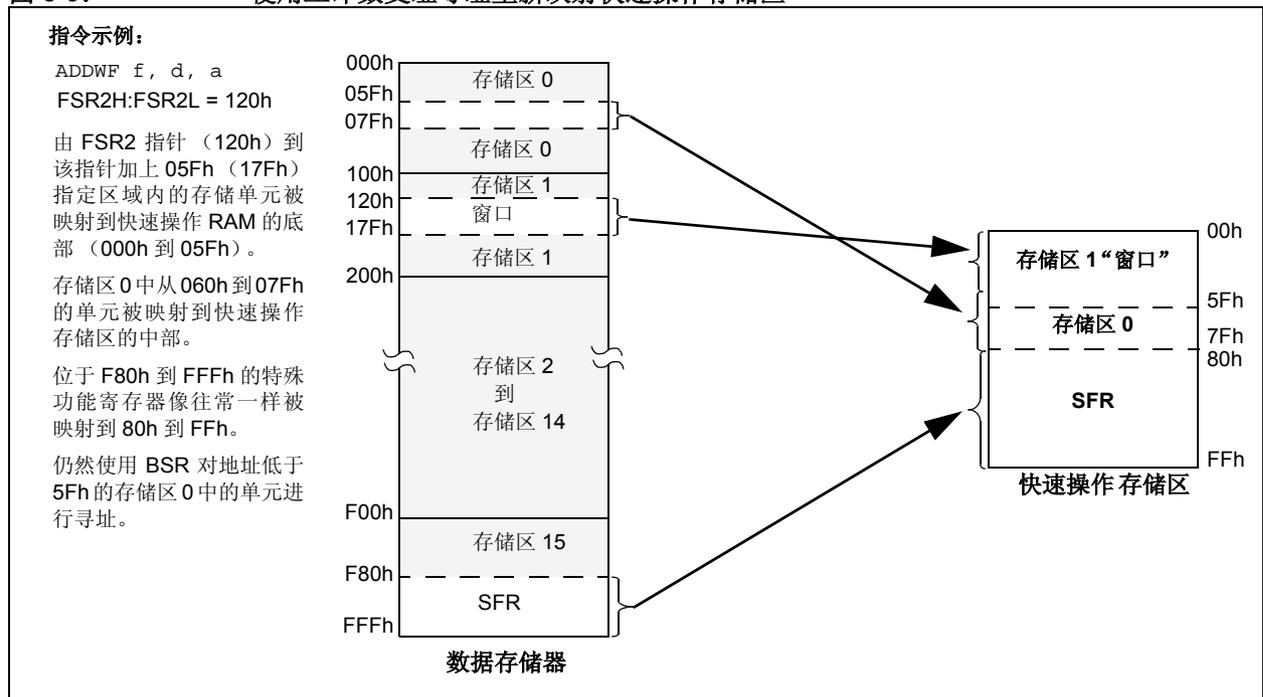
使用立即数变址寻址模式能改变快速操作 RAM 前 96 个单元地址 (00h 到 5Fh) 的映射方式。此模式映射存储区 0 的内容和由用户定义的可以位于数据存储器空间中任何地方的“窗口”内容，而不是仅仅包含存储区 0 下半部分的内容。FSR2 的值定义映射到窗口的地址的下边界，而上边界则由 FSR2 加 95 (5Fh) 决定。地址为 5Fh 以上的快速操作 RAM 的映射方法如前所述 (参见第 5.3.2 节“快速操作存储区”)。图 5-9 给出了在此寻址模式中重新映射快速操作存储区的示例。

快速操作存储区的重新映射 *仅适用于* 立即数变址寻址模式。使用 BSR (快速操作 RAM 位为 1) 的操作和前面一样继续使用直接寻址模式。

5.6 PIC18 指令执行和扩展指令集

使能扩展指令集会在现有的 PIC18 指令集中添加 8 条额外的指令。第 21.2 节“扩展的指令集”给出了这些指令执行的方式。

图 5-9: 使用立即数变址寻址重新映射快速操作存储区



PIC18F45J10 系列

注:

6.0 闪存程序存储器

在整个 VDD 范围内，闪存程序存储器在正常工作状态下都是可读写并可擦除的。

对程序存储器执行读操作时每次读取一个字节。对程序存储器执行写操作时每次写入一个 64 字节的数据块。对程序存储器执行擦除操作按照每次 1024 字节的数据块进行。不允许用户代码执行批量擦除操作。

写或擦除程序存储器将中止取指操作，直到写或擦除操作完成为止。在写或擦除期间不能访问程序存储器，因此无法执行代码。内部编程定时器可终止程序存储器的写入和擦除操作。

写入程序存储器的值不必是有效指令。执行存有无效指令的程序存储器单元会导致执行 NOP 指令。

6.1 表读和表写

为了读取和写入程序存储器，有两种操作可以让处理器在程序存储空间和数据 RAM 之间传送字节。

- 表读 (TBLRD)
- 表写 (TBLWT)

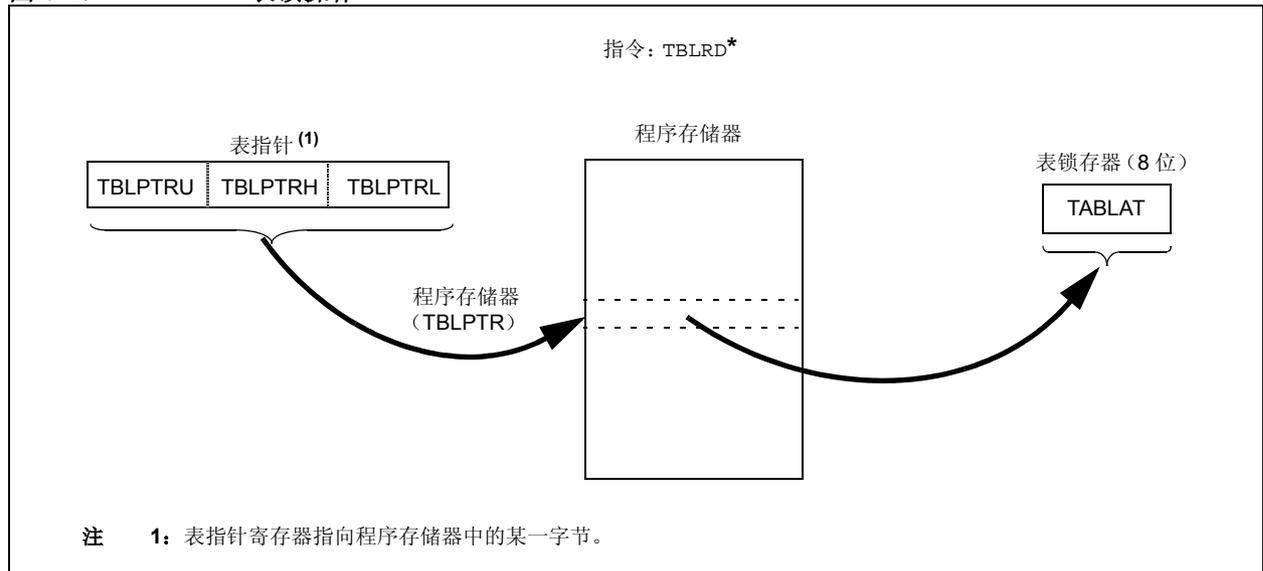
程序存储空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读和表写操作通过一个 8 位寄存器 (TABLAT) 在这两个存储空间之间传送数据。

表读操作从程序存储器获取数据并将其存入数据 RAM。图 6-1 给出了在程序存储器和数据 RAM 之间进行表读操作的过程。

表写操作将数据存储空间中的数据存入程序存储器中的保持寄存器。第 6.5 节“写入闪存程序存储器”将详细介绍将保持寄存器中的内容写入程序存储器的过程。图 6-2 给出了在程序存储器和数据 RAM 之间进行表写操作的过程。

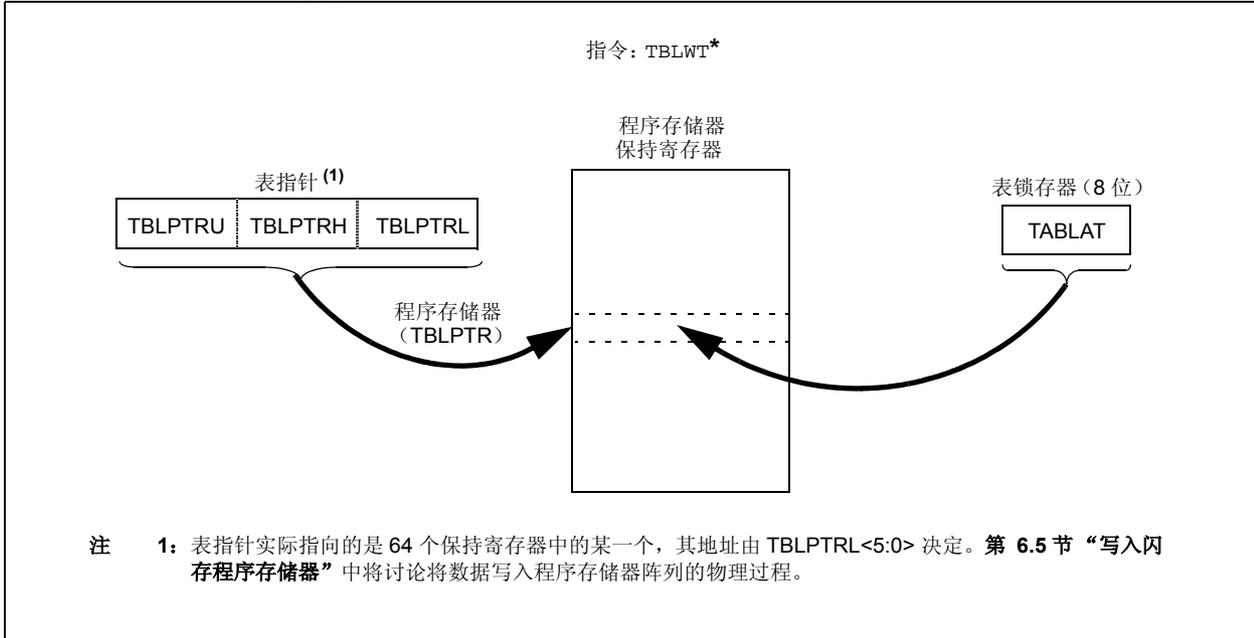
表操作以字节为单位进行。一个仅包含数据而非程序指令的表块不必字对齐。因此，表块可在任何字节地址处开始和结束。如果使用表写操作将可执行代码写入程序存储器，程序指令就需要进行字对齐。

图 6-1: 表读操作



PIC18F45J10 系列

图 6-2: 表写操作



6.2 控制寄存器

TBLRD 和 TBLWT 指令要用到几个控制寄存器。包括:

- EECON1 寄存器
- EECON2 寄存器
- TABLAT 寄存器
- TBLPTR 寄存器

6.2.1 EECON1 和 EECON2 寄存器

EECON1 寄存器 (寄存器 6-1) 是访问存储器的控制寄存器。EECON2 寄存器不是物理寄存器, 它专用于存储器的擦写操作。读取 EECON2 得到的是全 0。

若将 FREE 位置 1, 则允许对程序存储器进行擦除操作。擦除操作由下一个 WR 命令触发。当 FREE 位清零时, 则只使能写操作。

若将 WREN 位置 1, 则允许写操作。上电时将清零 WREN 位。在 WR 位置 1 时, WRERR 位将被硬件置 1; 当内部编程定时器超时并且写操作完成时, 清零 WRERR 位。

注: 如果在正常工作期间, WRERR 的读取值为 1, 则表明写操作因复位而提早终止或进行了非法的写操作。

控制位 WR 用于启动写操作。此位只能由软件置 1 而不能清零。写操作完成后, 由硬件将其清零。

寄存器 6-1:

EECON1: 数据 EEPROM 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0	
—	—	—	FREE	WRERR	WREN	WR	—	
bit 7								bit 0

bit 7-5 未用: 读为 0

bit 4 **FREE:** 闪存行擦除使能位

1 = 在下一个 WR 命令时, 擦除由 TBLPTR 指定的程序存储器行 (擦除操作完成时清零)
0 = 仅执行写操作

bit 3 **WRERR:** 闪存程序 / 数据 EEPROM 故障标志位

1 = 写操作提早终止 (由于正常工作中自定时编程期间的任何复位, 或非法写入)
0 = 写操作完成

bit 2 **WREN:** 闪存程序 / 数据 EEPROM 写使能位

1 = 允许闪存程序 / 数据 EEPROM 的写周期
0 = 禁止闪存程序 / 数据 EEPROM 的写周期

bit 1 **WR:** 写控制位

1 = 启动数据 EEPROM 擦写周期或程序存储器擦写周期。
(该操作是自定时的, 一旦写入完成即由硬件将该位清零。软件只能将该位置 1 而不能清零。)
0 = 写入 EEPROM 的周期完成

bit 0 未用: 读为 0

图注:

R = 可读位

W = 可写位

S = 只能由软件置 1 而不能清零的位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

6.2.2 TABLAT——表锁存寄存器

表锁存器 (Table Latch, TABLAT) 是映射到 SFR 空间的一个 8 位寄存器。它用于在程序存储器和数据 RAM 之间传输数据时保存 8 位数据。

6.2.3 TBLPTR——表指针寄存器

表指针 (Table Pointer, TBLPTR) 在程序存储器中寻址字节。TBLPTR 由 3 个 SFR 寄存器组成: 表指针最高字节、表指针高字节和表指针低字节 (TBLPTRU:TBLPTRH:TBLPTRL)。这 3 个寄存器合起来组成一个 22 位宽的指针, 其中低 21 位可使器件寻址至多 2 MB 的程序存储空间。第 22 位则允许访问器件 ID、用户 ID 以及配置位。

TBLRD 和 TBLWT 指令使用表指针寄存器 TBLPTR。利用表操作的四种方法之一, 这些指令可以更新 TBLPTR。表 6-1 列出了这些操作。这些操作只会影响 TBLPTR 的低 21 位。

6.2.4 表指针边界

TBLPTR 用于闪存程序存储器的读取、写入和擦除。

当执行 TBLRD 时, 表指针的所有 22 位决定将程序存储器哪个单元的数据读入 TABLAT。

当执行 TBLWT 时, 表指针寄存器的低 6 位 (TBLPTR<5:0>) 决定要写入 64 个程序存储器保持寄存器的哪一个。当开始定时写入程序存储器时 (通过 WR 位), 表指针寄存器的高 16 位 (TBLPTR<21:6>) 决定要写入哪一个 64 字节的程序存储块。如需更多详情, 请参见第 6.5 节 “写入闪存程序存储器”。

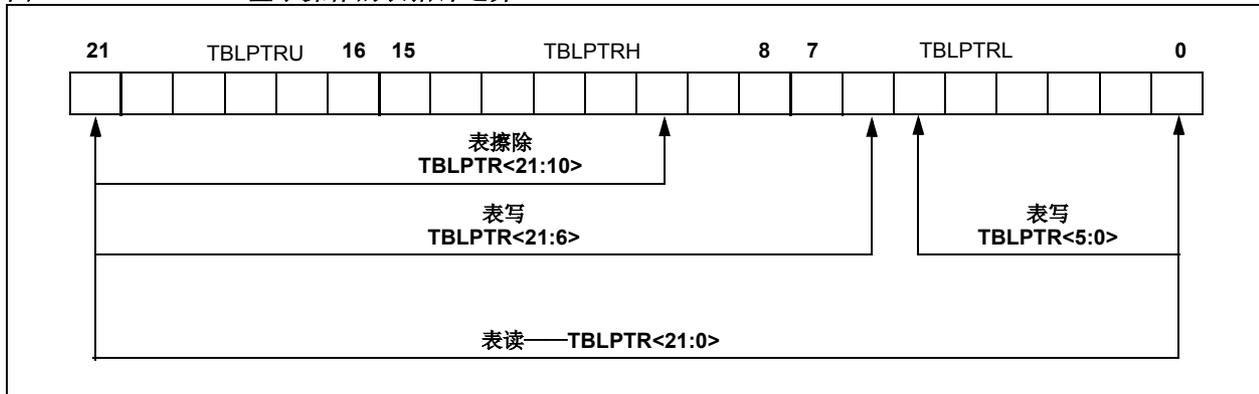
当擦除程序存储器时, 表指针的高 12 位 (TBLPTR<21:10>) 指向将要被擦除的 1024 字节块, 其低 10 位 (TBLPTR<9:0>) 被忽略。

图 6-3 说明了基于闪存程序存储器操作的相关 TBLPTR 边界。

表 6-1: 使用 TBLRD 和 TBLWT 指令执行表指针操作

示例	表指针操作
TBLRD* TBLWT*	不修改 TBLPTR
TBLRD** TBLWT**	TBLPTR 在读 / 写后递增
TBLRD*- TBLWT*-	TBLPTR 在读 / 写后递减
TBLRD+* TBLWT+*	TBLPTR 在读 / 写前递增

图 6-3: 基于操作的表指针边界



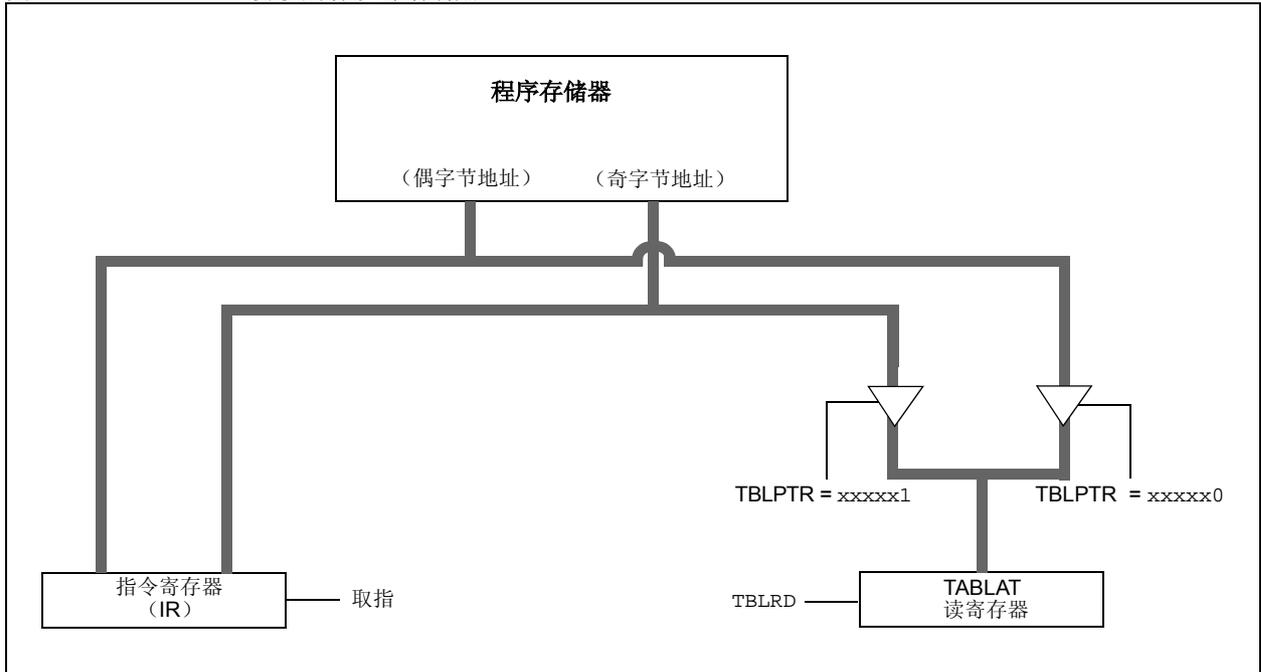
6.3 读取闪存程序存储器

TBLRD 指令用于从程序存储器获取数据并存入数据 RAM。表读操作每次从程序存储器读取一个字节。

TBLPTR 指向程序存储空间内的某个字节。执行 TBLRD 会将指向的字节存入 TABLAT。另外，可以自动修改 TBLPTR 以进行下次表读操作。

内部程序存储器通常是以字为单位构成的。由地址的最低有效位选择字的高字节或者低字节。图 6-4 显示了内部程序存储器和 TABLAT 之间的接口。

图 6-4: 读取闪存程序存储器



例 6-1: 读取一个闪存程序存储器字

```

MOVLW    CODE_ADDR_UPPER      ; Load TBLPTR with the base
MOVWF    TBLPTRU              ; address of the word
MOVLW    CODE_ADDR_HIGH
MOVWF    TBLPTRH
MOVLW    CODE_ADDR_LOW
MOVWF    TBLPTRL

READ_WORD
TBLRD*+                ; read into TABLAT and increment
MOVF     TABLAT, W       ; get data
MOVWF    WORD_EVEN
TBLRD*+                ; read into TABLAT and increment
MOVFW   TABLAT, W       ; get data
MOVF     WORD_ODD
    
```

PIC18F45J10 系列

6.4 擦除闪存程序存储器

最小擦除块为 1024 个字节。只有通过使用外部编程器或通过 ICSP 控制，才能够批量擦除更大的程序存储器块。闪存阵列不支持字擦除。

当从单片机本身启动擦除过程时，将擦除程序存储器的一个 1024 字节的块。TBLPTR<21:10> 指向将被擦除的块。TBLPTR<9:0> 被忽略。

擦除操作由 EECON1 寄存器控制。必须将 WREN 位置 1 以使能写操作。必须将 FREE 位置 1 以选择擦除操作。

作为保护机制，必须采用 EECON2 的写操作启动序列。

对于擦除内部闪存，长写周期是必要的。在长写周期中，指令暂停执行。内部编程定时器将终止长写周期操作。

6.4.1 闪存程序存储器擦除序列

擦除内部程序存储器块的事件顺序如下：

1. 将要擦除的行地址装入表指针。
2. 设置 EECON1 寄存器以执行擦除操作：
 - 将 WREN 位置 1 以使能写操作；
 - 将 FREE 位置 1 以使能擦除操作。
3. 禁止中断。
4. 向 EECON2 写入 55h。
5. 向 EECON2 写入 0AAh。
6. 将 WR 位置 1。这将启动行擦除周期。
7. 在擦除操作期间，CPU 将停止工作（内部定时器计时 2 ms 左右）。
8. 重新允许中断。

例 6-2: 擦除闪存程序存储器行

```
MOV LW CODE_ADDR_UPPER           ; load TBLPTR with the base
MOV WF TBLPTRU                   ; address of the memory block
MOV LW CODE_ADDR_HIGH
MOV WF TBLPTRH
MOV LW CODE_ADDR_LOW
MOV WF TBLPTRL

ERASE_ROW
BSF EECON1, WREN                 ; enable write to memory
BSF EECON1, FREE                 ; enable Row Erase operation
BCF INTCON, GIE                 ; disable interrupts

必须的序列
MOV LW 55h                       ; write 55h
MOV WF EECON2
MOV LW 0AAh
MOV WF EECON2                     ; write 0AAh
BSF EECON1, WR                   ; start erase (CPU stall)
BSF INTCON, GIE                 ; re-enable interrupts
```

6.5 写入闪存程序存储器

最小编程块为 32 字（即 64 字节）。不支持字或字节编程。

表写操作用于将数据装入编程闪存存储器所需的保持寄存器。表写操作使用 64 个保持寄存器进行编程。

因为表锁存器（TABLAT）仅一个单字节大小，所以每次编程操作必须执行 64 次 TBLWT 指令。由于只写入保持寄存器，所有表写操作实际上都是短周期写操作。在更新完 64 个寄存器后，必须写入 EECON1 寄存器，以便使用长写操作开始编程。

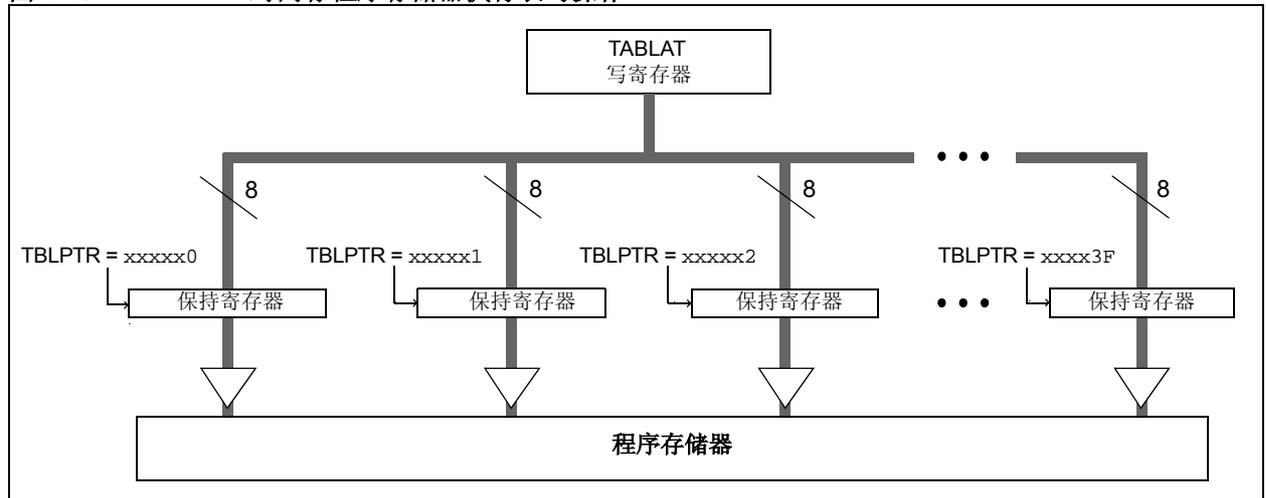
对于编程内部闪存而言，长写周期是必须的。在长写周期中，指令暂停执行。内部编程定时器将终止长写周期操作。

EEPROM 片上定时器控制写操作的时间。写 / 擦除电压可在器件电压范围内运行的片上电荷泵产生。

注： 与早期的器件不同，PIC18F45J10 系列器件在写操作发生后并不会复位保持寄存器。必须在编程序列之前清零或改写保持寄存器。

为了保证闪存的耐擦写寿命，在两次擦除操作之间不应对其某一闪存字节编程多于两次。在第三次尝试改写目标行的内容之前，需要对其执行批量或行擦除。

图 6-5: 对闪存程序存储器执行表写操作



6.5.1 闪存程序存储器写操作序列

对内部程序存储器编程的事件顺序应该为：

1. 如果要写入的程序存储器段已被事先编程，那么在写入之前必须首先将其擦除（见第 6.4.1 节“闪存程序存储器擦除序列”）。
2. 通过自动递增将 64 个字节写入保持寄存器。
3. 设置 EECON1 寄存器以执行写操作：
 - 将 WREN 位置 1 以使能字节写入。
4. 禁止中断。

5. 向 EECON2 写入 55h。
6. 向 EECON2 写入 0AAh。
7. 将 WR 位置 1。这将启动写周期。
8. 在写操作期间 CPU 将停止工作（内部定时器计时 2 ms 左右）。
9. 重新允许中断。
10. 校验存储器（表读）。

例 6-3 给出了所需的代码示例。

注： 在 WR 位置 1 前，表指针地址需要在保持寄存器的 64 字节地址范围内。

PIC18F45J10 系列

例 6-3: 写入闪存程序存储器

```

                                MOVLW   CODE_ADDR_UPPER           ; Load TBLPTR with the base
                                MOVWF   TBLPTRU                 ; address of the memory block
                                MOVLW   CODE_ADDR_HIGH
                                MOVWF   TBLPTRH
                                MOVLW   CODE_ADDR_LOW
                                MOVWF   TBLPTRL

ERASE_BLOCK

                                BSF     EECON1, WREN             ; enable write to memory
                                BSF     EECON1, FREE           ; enable Row Erase operation
                                BCF     INTCON, GIE             ; disable interrupts
                                MOVLW   55h
                                MOVWF   EECON2                 ; write 55h
                                MOVLW   0AAh
                                MOVWF   EECON2                 ; write 0AAh
                                BSF     EECON1, WR             ; start erase (CPU stall)
                                BSF     INTCON, GIE             ; re-enable interrupts
                                MOVLW   D'16'
                                MOVWF   WRITE_COUNTER           ; Need to write 16 blocks of 64 to write
                                                                ; one erase block of 1024

RESTART_BUFFER

                                MOVLW   D'64'
                                MOVWF   COUNTER
                                MOVLW   BUFFER_ADDR_HIGH       ; point to buffer
                                MOVWF   FSR0H
                                MOVLW   BUFFER_ADDR_LOW
                                MOVWF   FSR0L

FILL_BUFFER

                                ...                               ; read the new data from I2C, SPI,
                                                                ; PSP, USART, etc.

WRITE_BUFFER

                                MOVLW   D'64'                   ; number of bytes in holding register
                                MOVWF   COUNTER

WRITE_BYTE_TO_HREGS

                                MOVFF   POSTINC0, WREG           ; get low byte of buffer data
                                MOVWF   TABLAT                 ; present data to table latch
                                TBLWT+*                       ; write data, perform a short write
                                                                ; to internal TBLWT holding register.
                                                                ; loop until buffers are full

                                DECFSZ  COUNTER
                                BRA     WRITE_WORD_TO_HREGS

PROGRAM_MEMORY

                                BSF     EECON1, WREN             ; enable write to memory
                                BCF     INTCON, GIE             ; disable interrupts
                                MOVLW   55h
                                MOVWF   EECON2                 ; write 55h
                                MOVLW   0AAh
                                MOVWF   EECON2                 ; write 0AAh
                                BSF     EECON1, WR             ; start program (CPU stall)
                                BSF     INTCON, GIE             ; re-enable interrupts
                                BCF     EECON1, WREN             ; disable write to memory

                                DECFSZ  WRITE_COUNTER           ; done with one write cycle
                                BRA     RESTART_BUFFER           ; if not done replacing the erase block
```

必需的
序列

6.5.2 写校验

根据具体的应用，好的编程习惯一般要求使用原始值对写入值进行校验。当连续写入过多的数据已接近规范极限值时，就应该采用写校验。

6.5.3 写操作意外终止

如果由于意外事件（如掉电或意外复位）终止了写操作，就应该对刚刚编程的存储单元进行验证，如有必要，还要重新进行编程。如果写入操作在正常操作过程中被 MCLR 复位或 WDT 超时复位中断，用户就可以根据需要检测 WRERR 位并重新写入。

6.5.4 避免误写操作

为了防止误写闪存程序存储器，必须遵循写操作启动顺序。更多详情，请参见第 20.0 节“CPU 的特殊性能”。

6.6 代码保护时的闪存程序存储器操作

如需了解有关闪存程序存储器代码保护的详情，请参见第 20.6 节“程序校验和代码保护”。

表 6-2: 与闪存程序存储器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 (TBLPTR<20:16>)					43
TBPLTRH	程序存储器表指针高字节 (TBLPTR<15:8>)								43
TBLPTRL	程序存储器表指针低字节 (TBLPTR<7:0>)								43
TABLAT	程序存储器表锁存器								43
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
EECON2	EEPROM 控制寄存器 2 (非物理寄存器)								45
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45

图注: — = 未用，读为 0。访问闪存 /EEPROM 存储器时不使用阴影单元。

PIC18F45J10 系列

注:

7.0 8 x 8 硬件乘法器

7.1 简介

所有的 PIC18 器件均包含一个 8 x 8 硬件乘法器（乘法器是 ALU 的一部分）。该乘法器可执行无符号运算并产生一个 16 位运算结果，该结果存储在乘积寄存器 PRODH:PRODL 中。该乘法器执行的运算不会影响 STATUS 寄存器中的任何标志位。

通过硬件执行乘法运算只需要 1 个指令周期。硬件乘法器具有更高的计算吞吐量并减少了乘法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 PIC18 器件。表 7-1 给出了各种硬件和软件乘法运算的比较，包括所需的存储空间和执行时间。

7.2 工作原理

例 7-1 给出了一个 8 x 8 无符号乘法运算的指令序列。当已在 WREG 寄存器中装入了一个乘数时，实现该运算仅需一条指令。

例 7-2 给出了执行 8 x 8 有符号乘法运算的指令序列。要弄清乘数的符号位，必须检查每个乘数的最高有效位（MSb），并做相应的减法。

例 7-1: 8 x 8 无符号乘法程序

```
MOVWF ARG1, W ;
MULWF ARG2 ; ARG1 * ARG2 ->
; PRODH:PRODL
```

例 7-2: 8 x 8 有符号乘法程序

```
MOVWF ARG1, W
MULWF ARG2 ; ARG1 * ARG2 ->
; PRODH:PRODL

BTFSC ARG2, SB ; Test Sign Bit
SUBWF PRODH, F ; PRODH = PRODH
; - ARG1

MOVWF ARG2, W
BTFSC ARG1, SB ; Test Sign Bit
SUBWF PRODH, F ; PRODH = PRODH
; - ARG2
```

表 7-1: 各种乘法运算的性能比较

程序	乘法实现方法	程序存储空间 (字数)	周期数 (最多)	时间		
				@40 MHz	@10 MHz	@4 MHz
8 x 8 无符号	软件乘法	13	69	6.9 μs	27.6 μs	69 μs
	硬件乘法	1	1	100 ns	400 ns	1 μs
8 x 8 有符号	软件乘法	33	91	9.1 μs	36.4 μs	91 μs
	硬件乘法	6	6	600 ns	2.4 μs	6 μs
16 x 16 无符号	软件乘法	21	242	24.2 μs	96.8 μs	242 μs
	硬件乘法	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 有符号	软件乘法	52	254	25.4 μs	102.6 μs	254 μs
	硬件乘法	35	40	4.0 μs	16.0 μs	40 μs

PIC18F45J10 系列

例 7-3 给出了 16 x 16 无符号乘法运算的指令序列。公式 7-1 为所使用的算法。32 位运算结果存储在 4 个寄存器 (RES3:RES0) 中。

公式 7-1: 16 x 16 无符号乘法算法

$$\begin{aligned} \text{RES3:RES0} &= \text{ARG1H:ARG1L} \cdot \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \cdot \text{ARG2H} \cdot 2^{16}) + \\ &\quad (\text{ARG1H} \cdot \text{ARG2L} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2H} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2L}) \end{aligned}$$

例 7-3: 16 x 16 无符号乘法程序

```

MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L->
                      ; PRODH:PRODL

;
MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;

;
MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H->
                      ; PRODH:PRODL

MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;

;
MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

;
MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F
    
```

例 7-4 给出了 16 x 16 有符号乘法运算的指令序列。公式 7-2 为所使用的算法。32 位运算结果存储在 4 个寄存器 (RES3:RES0) 中。要弄清乘数的符号位，必须检查每个乘数的最高有效位 (MSb)，并做相应的减法。

公式 7-2: 16 x 16 有符号乘法算法

$$\begin{aligned} \text{RES3:RES0} &= \text{ARG1H:ARG1L} \cdot \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \cdot \text{ARG2H} \cdot 2^{16}) + \\ &\quad (\text{ARG1H} \cdot \text{ARG2L} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2H} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2L}) + \\ &\quad (-1 \cdot \text{ARG2H} < 7 > \cdot \text{ARG1H:ARG1L} \cdot 2^{16}) + \\ &\quad (-1 \cdot \text{ARG1H} < 7 > \cdot \text{ARG2H:ARG2L} \cdot 2^{16}) \end{aligned}$$

例 7-4: 16 x 16 有符号乘法程序

```

MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                      ; PRODH:PRODL

MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;

;
MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                      ; PRODH:PRODL

MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;

;
MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

;
MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

;
BTSS ARG2H, 7        ; ARG2H:ARG2L neg?
BRA SIGN_ARG1        ; no, check ARG1
MOVF ARG1L, W
SUBWF RES2
MOVF ARG1H, W
SUBWFB RES3

;
SIGN_ARG1
BTSS ARG1H, 7        ; ARG1H:ARG1L neg?
BRA CONT_CODE        ; no, done
MOVF ARG2L, W
SUBWF RES2
MOVF ARG2H, W
SUBWFB RES3

;
CONT_CODE
    
```

8.0 中断

PIC18F45J10 系列的器件具有多个中断源及一个中断优先级功能，可以给大多数中断源分配高优先级或者低优先级。高优先级中断矢量地址为 0008h，低优先级中断矢量地址为 0018h。高优先级中断事件将中断所有正在处理的低优先级中断。

有 13 个寄存器用于控制中断操作。这些寄存器是：

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1、PIR2 和 PIR3
- PIE1、PIE2 和 PIE3
- IPR1、IPR2 和 IPR3

建议使用 MPLAB® IDE 提供的 Microchip 头文件命名这些寄存器中的位。这使得汇编器 / 编译器能够自动识别指定寄存器内这些位的位置。

通常，用三个位来控制中断源的操作。它们是：

- 标志位表明发生了中断事件
- 使能位当标志位置 1 时，使程序跳转到中断矢量地址处执行
- 优先级位用于选择是高优先级还是低优先级

通过将 IPEN 位 (RCON<7>) 置 1，可使能中断优先级功能。当使能中断优先级时，有 2 位可允许全局中断。将 GIEH 位 (INTCON<7>) 置 1，可允许所有优先级位已置 1 (高优先级) 的中断。将 GIEL 位 (INTCON<6>) 置 1，可允许所有优先级位已清零 (低优先级) 的中断。当中断标志位、使能位以及相应的全局中断允许位均被置 1 时，程序将立即跳转到中断矢量地址 0008h 或 0018h，跳转到哪个地址取决于优先级位的设置。通过设置相应的使能位可以禁止单个中断。

当 IPEN 位清零 (默认状态) 时，便会禁止中断优先级功能，并且中断与 PICmicro® 中档器件相兼容。在兼容模式下，各个中断源的中断优先级位均不起作用。INTCON<6> 是 PEIE 位，它可允许 / 禁止所有的外设中断源。INTCON<7> 是 GIE 位，它可允许 / 禁止所有的中断源。在兼容模式下，所有中断均跳转到地址 0008h。

当响应中断时，全局中断允许位被清零以禁止其他中断。如果清零 IPEN 位，全局中断允许位就是 GIE 位。如果使用中断优先级，这个位将是 GIEH 位或者 GIEL 位。高优先级中断源会中断低优先级中断。处理高优先级中断时，低优先级中断将不被响应。

返回地址被压入堆栈，PC 中装入中断矢量地址 (0008h 或 0018h)。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

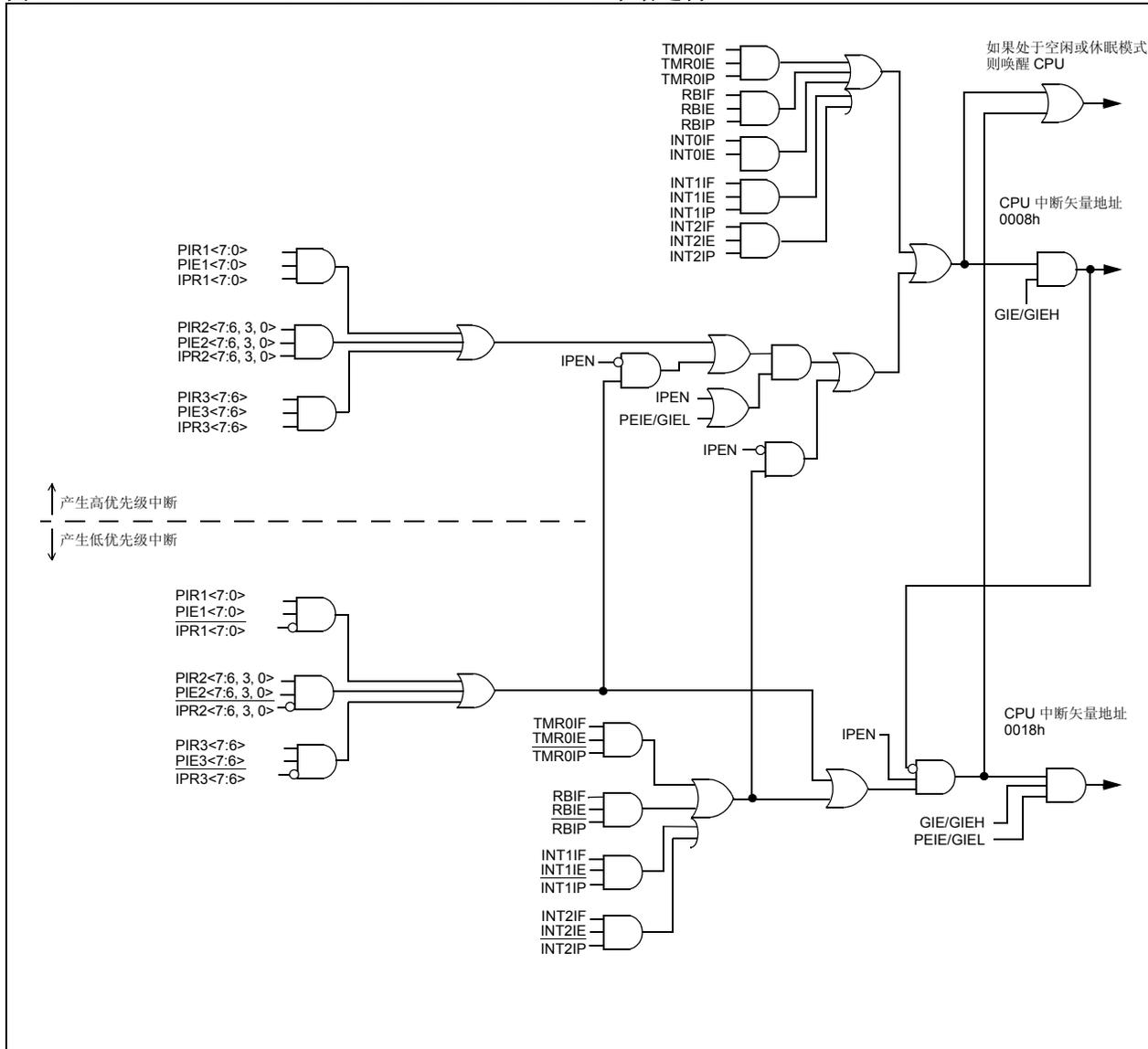
执行“从中断返回”指令 RETFIE 退出中断程序并将 GIE 位 (若使用中断优先级，则为 GIEH 或 GIEL 位) 置 1，从而重新允许中断。

对于外部中断事件，诸如 INT 引脚中断或者 PORTB 输入电平变化中断，中断响应延时将会是 3 到 4 个指令周期。对于单周期或双周期指令，中断响应延时完全相同。各中断标志位的置 1 不受对应的中断允许位和 GIE 位状态的影响。

注： 当允许任何中断时，不要使用 MOVFF 指令来修改任何中断控制寄存器。否则可能引起单片机执行出错。

PIC18F45J10 系列

图 8-1: PIC18F24J10/25J10/44J10/45J10 中断逻辑



8.1 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含多个使能位、优先级位和标志位。

注： 当一个中断发生时，不管对应的中断允许位或全局中断允许位的状态如何，中断标志位都将置 1。用户软件应在允许一个中断之前，确保先将相应的中断标志位清零。中断标志位可以用软件查询。

寄存器 8-1: INTCON: 中断控制寄存器

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7								bit 0

bit 7 **GIE/GIEH:** 全局中断允许位

当 IPEN = 0 时:

1 = 允许所有未屏蔽的中断

0 = 禁止所有中断

当 IPEN = 1 时:

1 = 允许所有高优先级中断

0 = 禁止所有中断

bit 6 **PEIE/GIEL:** 外设中断允许位

当 IPEN = 0 时:

1 = 允许所有未屏蔽的外设中断

0 = 禁止所有外设中断

当 IPEN = 1 时:

1 = 允许所有低优先级的外设中断

0 = 禁止所有低优先级的外设中断

bit 5 **TMR0IE:** TMR0 溢出中断允许位

1 = 允许 TMR0 溢出中断

0 = 禁止 TMR0 溢出中断

bit 4 **INT0IE:** INT0 外部中断允许位

1 = 允许 INT0 外部中断

0 = 禁止 INT0 外部中断

bit 3 **RBIE:** RB 端口电平变化中断允许位

1 = 允许 RB 端口电平变化中断

0 = 禁止 RB 端口电平变化中断

bit 2 **TMR0IF:** TMR0 溢出中断标志位

1 = TMR0 寄存器已经溢出 (必须由软件清零)

0 = TMR0 寄存器未溢出

bit 1 **INT0IF:** INT0 外部中断标志位

1 = 发生了 INT0 外部中断 (必须由软件清零)

0 = 未发生 INT0 外部中断

bit 0 **RBIF:** RB 端口电平变化中断标志位

1 = RB7:RB4 引脚中至少有一个发生了状态改变 (必须由软件清零)

0 = RB7:RB4 引脚的状态没有改变

注： 引脚上电平不匹配的情况会一直不断地将此位置 1。而对 PORTB 进行读操作，将结束该中断条件，并允许该位清零。

图注:

R = 可读位

W = 可写位

U = 未用位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

寄存器 8-2:

INTCON2: 中断控制寄存器 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEG0	INTEG1	INTEG2	—	TMR0IP	—	RBIP
bit 7							bit 0

- bit 7 **$\overline{\text{RBPU}}$** : PORTB 上拉使能位
1 = 禁止所有 PORTB 上拉
0 = 按各个端口锁存值使能 PORTB 上拉
- bit 6 **INTEG0**: 外部中断 0 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 5 **INTEG1**: 外部中断 1 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 4 **INTEG2**: 外部中断 2 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 3 未用: 读为 0
- bit 2 **TMR0IP**: TMR0 溢出中断优先级位
1 = 高优先级
0 = 低优先级
- bit 1 未用: 读为 0
- bit 0 **RBIP**: RB 端口电平变化中断优先级位
1 = 高优先级
0 = 低优先级

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

注: 当一个中断发生时, 不管对应的中断允许位或全局中断允许位的状态如何, 中断标志位都将置 1。用户软件应在允许一个中断之前, 确保先将相应的中断标志位清零。中断标志位可以用软件查询。

寄存器 8-3: INTCON3: 中断控制寄存器 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	
bit 7								bit 0

- bit 7 **INT2IP:** INT2 外部中断优先级位
1 = 高优先级
0 = 低优先级
- bit 6 **INT1IP:** INT1 外部中断优先级位
1 = 高优先级
0 = 低优先级
- bit 5 **未用:** 读为 0
- bit 4 **INT2IE:** INT2 外部中断允许位
1 = 允许 INT2 外部中断
0 = 禁止 INT2 外部中断
- bit 3 **INT1IE:** INT1 外部中断允许位
1 = 允许 INT1 外部中断
0 = 禁止 INT1 外部中断
- bit 2 **未用:** 读为 0
- bit 1 **INT2IF:** INT2 外部中断标志位
1 = 发生了 INT2 外部中断 (必须由软件清零)
0 = 未发生 INT2 外部中断
- bit 0 **INT1IF:** INT1 外部中断标志位
1 = 发生了 INT1 外部中断 (必须由软件清零)
0 = 未发生 INT1 外部中断

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

注: 当一个中断发生时, 不管对应的中断允许位或全局中断允许位的状态如何, 中断标志位都将置 1。用户软件应在允许一个中断之前, 确保先将相应的中断标志位清零。中断标志位可以用软件查询。

PIC18F45J10 系列

8.2 PIR 寄存器

PIR 寄存器包含各外设中断的标志位。根据外设中断源的数量，有 3 个外设中断请求（标志）寄存器（PIR1、PIR2 和 PIR3）。

注 1: 当有中断条件产生时，不管对应的中断允许位或全局允许位 GIE（INTCON<7>）的状态如何，中断标志位都将置 1。

2: 用户软件应在允许一个中断之前，确保先将相应的中断标志位清零；同时在响应该中断后，也应该将相应的中断标志位清零。

寄存器 8-4: PIR1: 外设中断请求（标志）寄存器 1

	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
bit 7								bit 0

- bit 7 **PSPIF:** 并行从动端口读 / 写中断标志位 ⁽¹⁾
 1 = 发生了读 / 写操作（必须由软件清零）
 0 = 未发生读 / 写操作
注: 在 28 引脚器件中不存在此位，应该被读为 0。
- bit 6 **ADIF:** A/D 转换器中断标志位
 1 = 完成 A/D 转换（必须由软件清零）
 0 = A/D 转换未完成
- bit 5 **RCIF:** EUSART 接收中断标志位
 1 = EUSART 接收缓冲器 RCREG 已满（当读取 RCREG 时清零）
 0 = EUSART 接收缓冲器为空
- bit 4 **TXIF:** EUSART 发送中断标志位
 1 = EUSART 发送缓冲器 TXREG 为空（当写入 TXREG 时清零）
 0 = EUSART 发送缓冲器已满
- bit 3 **SSP1IF:** 主控同步串行端口 1 中断标志位
 1 = 完成发送 / 接收（必须由软件清零）
 0 = 等待发送 / 接收
- bit 2 **CCP1IF:** ECCP1/CCP1 中断标志位
捕捉模式:
 1 = 发生了 TMR1 寄存器捕捉（必须由软件清零）
 0 = 未发生 TMR1 寄存器捕捉
比较模式:
 1 = 发生了与 TMR1 寄存器的比较匹配（必须由软件清零）
 0 = 未发生与 TMR1 寄存器的比较匹配
PWM 模式:
 在此模式下未使用。
- bit 1 **TMR2IF:** TMR2 与 PR2 匹配中断标志位
 1 = TMR2 与 PR2 发生了匹配（必须由软件清零）
 0 = TMR2 与 PR2 未发生匹配
- bit 0 **TMR1IF:** TMR1 溢出中断标志位
 1 = TMR1 寄存器溢出（必须由软件清零）
 0 = TMR1 寄存器未溢出

图注:

R = 可读位	W = 可写位	U = 未用位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

寄存器 8-5:

PIR2: 外设中断请求 (标志) 寄存器 2

R/W-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0	R/W-0
OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF
bit 7				bit 0			

- bit 7 **OSCFIF:** 振荡器故障中断标志位
 1 = 器件振荡器发生故障, 改由 INTOSC 提供时钟 (必须由软件清零)
 0 = 器件时钟正常运行
- bit 6 **CMIF:** 比较器中断标志位
 1 = 比较器输入已改变 (必须由软件清零)
 0 = 比较器输入未改变
- bit 5-4 **未用:** 读为 0
- bit 3 **BCL1IF:** 总线冲突中断标志位 (MSSP1 模块)
 1 = 发生了总线冲突 (必须由软件清零)
 0 = 未发生总线冲突
- bit 2-1 **未用:** 读为 0
- bit 0 **CCP2IF:** CCP2 中断标志位
捕捉模式:
 1 = 发生了 TMR1 寄存器捕捉 (必须由软件清零)
 0 = 未发生 TMR1 寄存器捕捉
比较模式:
 1 = 发生了与 TMR1 寄存器比较匹配 (必须由软件清零)
 0 = 未发生与 TMR1 寄存器的比较匹配
PWM 模式:
 在此模式下未使用。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

寄存器 8-6:

PIR3: 外设中断请求 (标志) 寄存器 3

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
SSP2IF	BCL2IF	—	—	—	—	—	—
bit 7				bit 0			

- bit 7 **SSP2IF:** 主控同步串行端口 2 中断标志位
 1 = 完成发送 / 接收 (必须由软件清零)
 0 = 等待发送 / 接收
- bit 6 **BCL2IF:** 总线冲突中断标志位 (MSSP2 模块)
 1 = 发生了总线冲突 (必须由软件清零)
 0 = 未发生总线冲突
- bit 5-0 **未用:** 读为 0

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

8.3 PIE 寄存器

PIE 寄存器包含各外设中断的使能位。根据外设中断源的数量，有 3 个外设中断允许寄存器（PIE1、PIE2 和 PIE3）。当 IPEN = 0 时，要允许任何外设中断就必须将 PEIE 位置 1。

寄存器 8-7: PIE1: 外设中断允许寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE:** 并行从动端口读 / 写中断允许位 ⁽¹⁾

1 = 允许 PSP 读 / 写中断
0 = 禁止 PSP 读 / 写中断

注: 在 28 引脚器件中不存在此位，应该被读为 0。

bit 6 **ADIE:** A/D 转换器中断允许位

1 = 允许 A/D 中断
0 = 禁止 A/D 中断

bit 5 **RCIE:** EUSART 接收中断允许位

1 = 允许 EUSART 接收中断
0 = 禁止 EUSART 接收中断

bit 4 **TXIE:** EUSART 发送中断允许位

1 = 允许 EUSART 发送中断
0 = 禁止 EUSART 发送中断

bit 3 **SSP1IE:** 主控同步串行端口 1 中断允许位

1 = 允许 MSSP1 中断
0 = 禁止 MSSP1 中断

bit 2 **CCP1IF:** ECCP1/CCP1 中断允许位

1 = 允许 ECCP1/CCP1 中断
0 = 禁止 ECCP1/CCP1 中断

bit 1 **TMR2IE:** TMR2 与 PR2 匹配中断允许位

1 = 允许 TMR2 与 PR2 匹配中断
0 = 禁止 TMR2 与 PR2 匹配中断

bit 0 **TMR1IE:** TMR1 溢出中断允许位

1 = 允许 TMR1 溢出中断
0 = 禁止 TMR1 溢出中断

图注:

R = 可读位

W = 可写位

U = 未用位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 8-8:

PIE2: 外设中断允许寄存器 2

R/W-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0	R/W-0
OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE
bit 7				bit 0			

- bit 7 **OSCFIE:** 振荡器故障中断允许位
1 = 允许
0 = 禁止
- bit 6 **CMIE:** 比较器中断允许位
1 = 允许
0 = 禁止
- bit 5-4 **未用:** 读为 0
- bit 3 **BCL1IF:** 总线冲突中断允许位 (MSSP1 模块)
1 = 允许
0 = 禁止
- bit 2-1 **未用:** 读为 0
- bit 0 **CCP2IE:** CCP2 中断允许位
1 = 允许
0 = 禁止

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

寄存器 8-9:

PIE3: 外设中断允许寄存器 3

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
SSP2IE	BCL2IE	—	—	—	—	—	—
bit 7				bit 0			

- bit 7 **SSP2IE:** 主控同步串行端口 2 中断允许位
1 = 允许
0 = 禁止
- bit 6 **BCL2IE:** 总线冲突中断允许位 (MSSP2 模块)
1 = 允许
0 = 禁止
- bit 5-0 **未用:** 读为 0

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

8.4 IPR 寄存器

IPR 寄存器包含各外设中断的优先级位。根据外设中断源的数量，有 3 个外设中断优先级寄存器（IPR1、IPR2 和 IPR3）。使用优先级位要求将中断优先级使能（IPEN）位置 1。

寄存器 8-10: IPR1: 外设中断优先级寄存器 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSP1P ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

bit 7 **PSP1P:** 并行从动端口读 / 写中断优先级位 ⁽¹⁾

- 1 = 高优先级
- 0 = 低优先级

注: 在 28 引脚器件中不存在此位，应该被读为 0。

bit 6 **ADIP:** A/D 转换器中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 5 **RCIP:** EUSART 接收中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 4 **TXIP:** EUSART 发送中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 3 **SSP1IP:** 主控同步串行端口 1 中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 2 **CCP1IP:** ECCP1/CCP1 中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 1 **TMR2IP:** TMR2 与 PR2 匹配中断优先级位

- 1 = 高优先级
- 0 = 低优先级

bit 0 **TMR1IP:** TMR1 溢出中断优先级位

- 1 = 高优先级
- 0 = 低优先级

图注:

R = 可读位

W = 可写位

U = 未用位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 8-11: IPR2: 外设中断优先级寄存器 2

R/W-1	R/W1	U-0	U-0	R/W-1	U-0	U-0	R/W-1
OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP
bit 7				bit 0			

- bit 7 **OSCFIP:** 振荡器故障中断优先级位
1 = 高优先级
0 = 低优先级
- bit 6 **CMIP:** 比较器中断优先级位
1 = 高优先级
0 = 低优先级
- bit 5-4 **未用:** 读为 0
- bit 3 **BCL1IP:** 总线冲突中断优先级位 (MSSP1 模块)
1 = 高优先级
0 = 低优先级
- bit 2-1 **未用:** 读为 0
- bit 0 **CCP2IP:** CCP2 中断优先级位
1 = 高优先级
0 = 低优先级

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

寄存器 8-12: IPR3: 外设中断优先级寄存器 3

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
SSP2IP	BCL2IP	—	—	—	—	—	—
bit 7				bit 0			

- bit 7 **SSP2IP:** 主控同步串行端口 2 中断优先级位
1 = 高优先级
0 = 低优先级
- bit 6 **BCL2IP:** 总线冲突中断优先级位 (MSSP2 模块)
1 = 高优先级
0 = 低优先级
- bit 5-0 **未用:** 读为 0

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

8.5 RCON 寄存器

RCON 寄存器包含几个位，可以用来确定器件上次复位或从空闲或休眠模式被唤醒的原因。RCON 还包含中断优先级使能位（IPEN）。

寄存器 8-13: RCON: 复位控制寄存器

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0	
IPEN	—	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}	
bit 7								bit 0

- bit 7 **IPEN:** 中断优先级使能位
1 = 使能中断优先级
0 = 禁止中断优先级（PIC16CXXX 兼容模式）
- bit 6-5 **未用:** 读为 0
- bit 4 **\overline{RI} :** RESET 指令标志位
关于位操作的详细信息，请参见寄存器 4-1。
- bit 3 **\overline{TO} :** 看门狗定时器超时溢出标志位
关于位操作的详细信息，请参见寄存器 4-1。
- bit 2 **\overline{PD} :** 掉电检测标志位
关于位操作的详细信息，请参见寄存器 4-1。
- bit 1 **\overline{POR} :** 上电复位状态位
关于位操作的详细信息，请参见寄存器 4-1。
- bit 0 **\overline{BOR} :** 欠压复位状态位
关于位操作的详细信息，请参见寄存器 4-1。

图注:

R = 可读位	W = 可写位	U = 未用位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

8.6 INTn 引脚中断

RB0/INT0、RB1/INT1 和 RB2/INT2 引脚上的外部中断都是通过边沿触发的。如果 INTCON2 寄存器中相应的 INTEDGx 位被置 1 (= 1)，则为上升沿触发；如果该位被清零，则为下降沿触发。当 RBx/INTx 引脚上出现一个有效边沿时，相应的标志位 INTxIF 被置 1。通过清零相应的允许位 INTxIE，可禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将中断标志位 INTxIF 清零。

如果 INTxIE 位在进入功耗管理模式前被置 1，则所有的外部中断 (INT0、INT1 和 INT2) 均能将处理器从功耗管理模式中唤醒。如果全局中断允许位 GIE 被置 1，则处理器将在被唤醒之后跳转到中断矢量处执行程序。

INT1 和 INT2 的中断优先级由中断优先级位 INT1IP (INTCON3<6>) 和 INT2IP (INTCON3<7>) 决定。没有与 INT0 相关的优先级位。INT0 始终是一个高优先级的中断源。

8.7 TMR0 中断

在 8 位模式 (默认模式) 下，TMR0 寄存器的溢出 (FFh → 00h) 会将标志位 TMR0IF 置 1。在 16 位模式下，TMR0H:TMR0L 寄存器对的溢出 (FFFFh → 0000h) 会将 TMR0IF 标志位置 1。通过将允许位 TMR0IE (INTCON<5>) 置 1 或清零，可以允许或禁止该中断。Timer0 的中断优先级由中断优先级位 TMR0IP (INTCON2<2>) 决定。欲进一步了解 Timer0 模块的详细内容，请参见第 10.0 节 “Timer0 模块”。

8.8 PORTB 电平变化中断

PORTB<7:4> 上的一个输入电平发生变化，会将标志位 RBIF (INTCON<0>) 置 1。通过将允许位 RBIE (INTCON<3>) 置 1 或清零，可以允许或禁止该中断。PORTB 电平变化中断的优先级由中断优先级位 RBIP (INTCON2<0>) 决定。

8.9 中断的现场保护

在中断期间，返回的 PC 地址被保存在堆栈中。另外，将 WREG、STATUS 以及 BSR 寄存器的值压入快速返回堆栈。如果未使用从中断快速返回功能 (见第 5.3 节 “数据存储寄存器构成”)，用户可能需要在进入中断服务程序前，保存 WREG、STATUS 以及 BSR 寄存器的值。根据用户的应用，还可能需要在保存其他寄存器的值。例 8-1 在执行中断服务程序期间，保存并恢复 WREG、STATUS 和 BSR 寄存器的值。

例 8-1: 将 STATUS、WREG 和 BSR 寄存器的值保存在 RAM 中

```

MOVWF    W_TEMP                ; W_TEMP is in virtual bank
MOVFF    STATUS, STATUS_TEMP    ; STATUS_TEMP located anywhere
MOVFF    BSR, BSR_TEMP          ; BSR_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF    BSR_TEMP, BSR          ; Restore BSR
MOVF     W_TEMP, W              ; Restore WREG
MOVFF    STATUS_TEMP, STATUS    ; Restore STATUS
    
```

PIC18F45J10 系列

注:

9.0 I/O 端口

根据选定的器件和使能的功能，最多有 5 个端口可用。I/O 端口的一些引脚与器件上的外设功能复用。通常，当外设使能时，其对应的引脚就不能被用作通用 I/O 引脚。

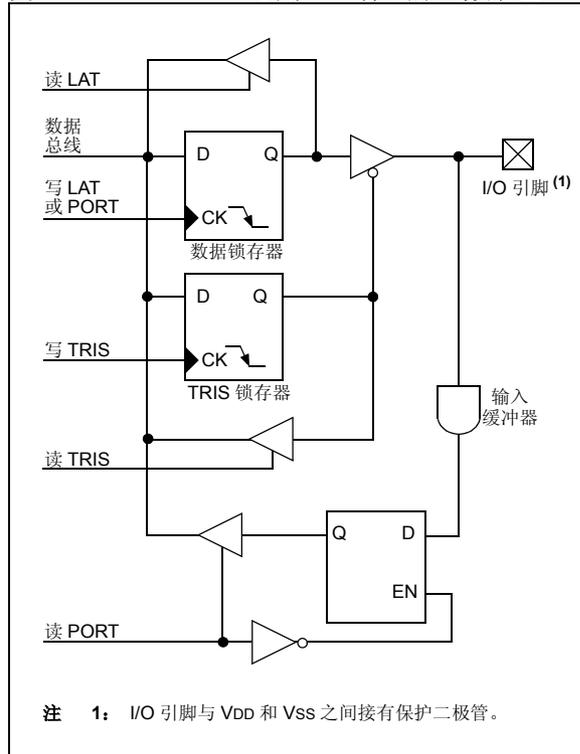
每个端口都有 3 个寄存器对其操作进行控制。它们是：

- TRIS 寄存器（数据方向寄存器）
- Port 寄存器（读取器件引脚的电平）
- LAT 寄存器（输出锁存器）

在对 I/O 引脚驱动值进行读 - 修改 - 写操作时会用到数据锁存器（LAT 寄存器）。

图 9-1 给出了不带外设接口的通用 I/O 端口的简化模型。

图 9-1: 通用 I/O 端口的工作原理



9.1 I/O 端口引脚能力

在应用开发设计时，必须考虑到端口引脚的能力。某些引脚上的输出驱动能力要高于其他引脚。同样，某些引脚可以接受高于 VDD 的输入电平。

9.1.1 引脚输出驱动能力

输出引脚的驱动能力各不相同，以满足各种应用的需求。PORTB 和 PORTC 是为驱动较高的负载，比如 LED 而设计的。其他端口都是为驱动较小的负载而设计的，通常只用于指示。表 9-1 汇总了输出驱动能力。更多详情，请参见第 23.0 节“电气规范”。

表 9-1: 输出驱动电平

端口	驱动	说明
PORTA	最低	用于指示。
PORTD		
PORTE		
PORTB	高	适合直接输出 LED 驱动电平。
PORTC		

9.1.2 输入引脚和电压注意事项

用作器件输入的引脚的电压容差取决于该引脚的输入功能。仅用作数字输入的引脚能够接受高达 5.5 V 的直流电压，即数字逻辑电路的典型电压值。相反，也具有模拟输入功能的引脚只可以接受最高为 VDD 的电压值。应尽量避免在这些引脚上施加超过 VDD 的电压。表 9-2 汇总了这些输入能力。更多详情，请参见第 23.0 节“电气规范”。

表 9-2: 输入电平

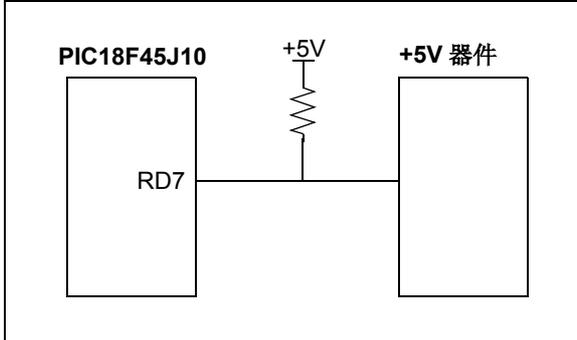
端口或引脚	可接受的最高输入电平	说明
PORTA<5:0>	VDD	只能接受小于 VDD 的输入电压。
PORTB<5:0>		
PORTC<1:0>		
PORTE<2:0>	5.5V	可接受的输入电压在 VDD 以上，对于大部分标准逻辑电路很有用。
PORTB<7:6>		
PORTC<7:2>		
PORTD<7:0>		

PIC18F45J10 系列

9.1.3 与 5V 系统接口

虽然 PIC18F45J10 系列器件的 VDDMAX 是 3.6V，但是即使目标系统的 VIH 高于 3.6V，这些器件仍然能够与 5V 系统相连。要实现这种接口，必须为引脚外接一个上拉电阻（图 9-2），清零该引脚的 LAT 位并设置对应的 TRIS 位（图 9-1）以允许将该引脚拉高或驱动为低电平。只有可接受最高为 5.5V 电压的端口引脚才能用于这种类型的接口（见第 9.1.2 节“输入引脚和电压注意事项”）。

图 9-2: +5V 系统硬件接口



例 9-1: 与 +5V 系统通信

```
BCF  LATD, 7 ; set up LAT register so
      ; changing TRIS bit will
      ; drive line low
BCF  TRISD, 7 ; send a 0 to the 5V system
BCF  TRISD, 7 ; send a 1 to the 5V system
```

9.2 PORTA、TRISA 和 LATA 寄存器

PORTA 是 5 位宽的双向端口。相应的数据方向寄存器为 TRISA。置位 TRISA (= 1) 可以让相应的 PORTA 引脚作为输入引脚（即将相应的输出驱动器置于高阻态模式）。清零 TRISA 位 (= 0) 将使相应的 PORTA 引脚作为输出引脚（即将输出锁存器的内容置于所选择的引脚上）。

读 PORTA 寄存器就是读引脚状态，而写该寄存器就是写入端口锁存器。

数据锁存器 (LATA) 也是映射的存储器。对 LATA 寄存器执行读—修改—写操作实际上是读写 PORTA 的输出锁存值。

其他 PORTA 引脚与模拟 VREF+ 和 VREF- 输入以及比较器参考电压输出引脚复用。通过清零或置位 ADCON1 寄存器 (A/D 控制寄存器 1) 中的控制位，将 RA3:RA0 和 RA5 引脚作为 A/D 转换器输入通道。

通过设置 CMCON 寄存器中的相应位，还可以将 RA0 和 RA3 引脚用作比较器的输入端，并将 RA5 用作 C2 比较器的输出端。要将 RA3:RA0 用作数字输入引脚，则必须关闭比较器。

注： 在上电复位时，RA5 和 RA3:RA0 被配置为模拟输入并读为 0。

所有的 PORTA 引脚都具有 TTL 输入电平和 CMOS 输出驱动器。

即使在 PORTA 引脚被用作模拟输入的时候，TRISA 寄存器仍然控制 PORTA 引脚的方向。在这种情况下，用户必须确保 TRISA 寄存器中的位保持为置 1 状态。

例 9-2: 初始化 PORTA

```
CLRF  PORTA ; Initialize PORTA by
           ; clearing output
           ; data latches
CLRF  LATA ; Alternate method
           ; to clear output
           ; data latches
MOVLW 07h ; Configure A/D
MOVWF  ADCON1 ; for digital inputs
MOVWF 07h ; Configure comparators
MOVWF CMCON ; for digital input
MOVLW 0CFh ; Value used to
           ; initialize data
           ; direction
MOVWF TRISA ; Set RA<3:0> as inputs
           ; RA<5:4> as outputs
```

表 9-3: PORTA I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RA0/AN0	RA0	0	O	DIG	LATA<0> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<0> 数据输入；使能模拟输入时被禁止。
	AN0	1	I	ANA	A/D 输入通道 0 和比较器 C1 - 输入。发生上电复位时的默认输入配置；不影响数字输出。
RA1/AN1	RA1	0	O	DIG	LATA<1> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<1> 数据输入；使能模拟输入时被禁止。
	AN1	1	I	ANA	A/D 输入通道 1 和比较器 C2 - 输入。发生上电复位时的默认输入配置；不影响数字输出。
RA2/AN2/ VREF-/CVREF	RA2	0	O	DIG	LATA<2> 数据输出；不受模拟输入影响。当使能 CVREF 输出时被禁止。
		1	I	TTL	PORTA<2> 数据输入。当使能模拟功能时被禁止；当使能 CVREF 输出时被禁止。
	AN2	1	I	ANA	A/D 输入通道 2 和比较器 C2+ 输入。发生上电复位时的默认输入配置；不受模拟输出的影响。
	VREF-	1	I	ANA	A/D 和比较器参考电压低电平输入。
	CVREF	x	O	ANA	比较器参考电压输出。使能此功能将禁止数字 I/O。
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<3> 数据输入；使能模拟输入时被禁止。
	AN3	1	I	ANA	A/D 输入通道 3 和比较器 C1+ 输入。发生上电复位时的默认输入配置。
	VREF+	1	I	ANA	A/D 和比较器参考电压高电平输入。
RA5/AN4/SS1/ C2OUT	RA5	0	O	DIG	LATA<5> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<5> 数据输入；使能模拟输入时被禁止。
	AN4	1	I	ANA	A/D 输入通道 4。发生上电复位时的默认配置。
	SS1	1	I	TTL	MSSP1 的从动选择输入（MSSP1 模块）。
	C2OUT	0	O	DIG	比较器 2 输出；优先级高于端口数据。
OSC2/CLKO	OSC2	x	O	ANA	主振荡器反馈输出连接（HS 模式）。
	CLKO	x	O	DIG	RC 和 EC 振荡器模式下的系统周期时钟输出（Fosc/4）。
OSC1/CLKI	OSC1	x	I	ANA	主振荡器输入连接。
	CLKI	x	I	ANA	主时钟输入连接。

图注： DIG = 数字电平输出； TTL = TTL 输入缓冲器； ST = 施密特触发器输入缓冲器； ANA = 模拟电平输入 / 输出； x = 任意值（TRIS 位不影响端口方向或者在此选项中被忽略）。

PIC18F45J10 系列

表 9-4: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTA	—	—	RA5	—	RA3	RA2	RA1	RA0	46
LATA	—	—	PORTA 数据锁存器 (读取和写入数据锁存器)						46
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	44
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	45
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	45

图注: — = 未用 (读为 0)。PORTA 未使用阴影单元。

9.3 PORTB、TRISB 和 LATB 寄存器

PORTB 是 8 位宽的双向端口。它对应的数据方向寄存器是 TRISB。置位 TRISB (= 1) 可以让相应 PORTB 引脚作为输入引脚（即将相应的输出驱动器置于高阻态模式）。清零 TRISB 位 (= 0) 将使相应的 PORTB 引脚作为输出引脚（即将输出锁存器的内容置于所选择的引脚上）。

数据锁存器（LATB）也是映射的存储器。对 LATB 寄存器执行读—修改—写操作实际上是读写 PORTB 的输出锁存值。

例 9-3: 初始化 PORTB

```

CLRF   PORTB    ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF   LATB     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0Fh     ; Set RB<4:0> as
MOVWF  ADCON1  ; digital I/O pins
                ; (required if config bit
                ; PBDEN is set)
MOVLW  0CFh   ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISB   ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
    
```

PORTB 的每个引脚都有内部弱上拉电路。单个控制位可以开启所有上拉电路。可以通过清零 RBPUR 位（INTCON2<7>）来开启上拉电路。当端口引脚配置为输出时，其弱上拉电路会自动切断。此弱上拉功能在上电复位时被禁止。

注： 在上电复位时，默认情况下 RB4:RB0 被配置为模拟输入并读为 0；RB7:RB5 被配置为数字输入。
通过对配置位 PBDEN 进行编程，在发生上电复位时，也可将 RB4:RB0 配置为数字输入引脚。

PORTB 的 4 个引脚（RB7:RB4）都有电平变化中断功能。只有配置为输入的引脚会导致中断发生（即，当 RB7:RB4 的任何一个引脚被配置为输出时，该引脚不再具有电平变化中断比较功能）。RB7:RB4 输入引脚上的电平与上次读 PORTB 时锁存的旧值进行比较。RB7:RB4 输出的“不匹配”值一起进行逻辑“或”运算，产生 RB 端口电平变化中断，并将标志位 RBIF（INTCON<0>）置 1。

此中断可将器件从休眠模式或任何空闲模式唤醒。用户可用以下方式在中断服务程序中清除该中断：

- a) 读或写 PORTB（MOVFF（ANY），PORTB 指令除外）。
- b) 清零标志位 RBIF。

电平不匹配的状态将会一直不断地将 RBIF 标志位置 1。而读 PORTB 将结束该状态，并允许将 RBIF 标志位清零。

建议使用此电平变化触发中断来实现按键唤醒以及其他仅使用 PORTB 该中断的应用。在使用电平变化触发中断功能时，建议不要查询 PORTB 的状态。

通过配置位 CCP2MX 可将 RB3 配置为 CCP2 模块的备用外设引脚（CCP2MX = 0）。

RB5 引脚与 Timer0 模块的时钟输入以及比较器的某个输出复用，成为 RB5/KBI1/T0CKI/C1OUT 引脚。

PIC18F45J10 系列

表 9-5: PORTB I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RB0/INT0/FLT0/ AN12	RB0	0	O	DIG	LATB<0> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<0> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	INT0	1	I	ST	外部中断 0 输入。
	FLT0	1	I	ST	PWM 错误输入（ECCP1/CCP1 模块）；用软件使能。
	AN12	1	I	ANA	A/D 输入通道 12。(1)
RB1/INT1/AN10	RB1	0	O	DIG	LATB<1> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<1> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	INT1	1	I	ST	外部中断 1 输入。
AN10	1	I	ANA	A/D 输入通道 10。(1)	
RB2/INT2/AN8	RB2	0	O	DIG	LATB<2> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<2> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	INT2	1	I	ST	外部中断 2 输入。
	AN8	1	I	ANA	A/D 输入通道 8。(1)
RB3/AN9/CCP2	RB3	0	O	DIG	LATB<3> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<3> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN9	1	I	ANA	A/D 输入通道 9。(1)
	CCP2(2)	0	O	DIG	CCP2 比较和 PWM 输出。
		1	I	ST	CCP2 捕捉输入。
RB4/KBI0/AN11	RB4	0	O	DIG	LATB<4> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<4> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	KBI0	1	I	TTL	电平变化中断引脚。
	AN11	1	I	ANA	A/D 输入通道 11。(1)
RB5/KBI1/T0CKI/ C1OUT	RB5	0	O	DIG	LATB<5> 数据输出。
		1	I	TTL	PORTB<5> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI1	1	I	TTL	电平变化中断引脚。
	T0CKI	1	I	ST	Timer0 时钟输入。
	C1OUT	0	O	DIG	比较器 1 输出；优先级高于端口数据。
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> 数据输出。
		1	I	TTL	PORTB<6> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI2	1	I	TTL	电平变化中断引脚。
	PGC	x	I	ST	ICSP 和 ICD 工作的串行时钟（ICSP™）输入。(3)
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> 数据输出。
		1	I	TTL	PORTB<7> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI3	1	I	TTL	电平变化中断引脚。
	PGD	x	O	DIG	执行 ICSP 和 ICD 操作时串行输出数据。(3)
		x	I	ST	执行 ICSP 和 ICD 操作时串行输入数据。(3)

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；x = 任意值（TRIS 位不影响端口方向或者在此选项中被忽略）。

- 注 1: 发生上电复位时的配置由 PBADEN 配置位决定。默认情况下（PBADEN 置 1），这些引脚被配置为模拟输入；而当 PBADEN 清零时，这些引脚被配置为数字输入。
- 2: 当 CCP2MX 配置位为 0 时，对 CCP2 进行其他分配。默认配置为 RC1。
- 3: 当使能 ICSP 或 ICD 时，禁止所有其他引脚功能。

表 9-6: 与 PORTB 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	46
LATB	PORTB 数据锁存器 (读取和写入数据锁存器)								46
TRISB	PORTB 数据方向控制寄存器								46
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	43
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	43
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	44

图注: — = 未用 (读为 0)。PORTB 未使用阴影单元。

PIC18F45J10 系列

9.4 PORTC、TRISC 和 LATC 寄存器

PORTC 是 8 位宽的双向端口。它对应的数据方向寄存器是 TRISC。置位 TRISC (= 1) 可以让相应 PORTC 引脚作为输入引脚（即将相应的输出驱动器置于高阻态模式）。清零 TRISC 位 (= 0) 将使相应的 PORTC 引脚作为输出引脚（即将输出锁存器的内容置于所选择的引脚上）。

数据锁存器（LATC）也是映射的存储器。对 LATC 寄存器执行读—修改—写操作实际上是读写 PORTC 的输出锁存值。

PORTC 与几种外设功能复用（表 9-7）。引脚有施密特触发输入缓冲器。通常由配置位 CCP2MX 将 RC1 配置为 CCP2 模块的默认外设引脚（默认 / 擦除状态，CCP2MX = 1）。

当使能外设功能时，要小心定义每个 PORTC 引脚的 TRIS 位。有些外设会改写相应引脚的 TRIS 位方向设置而将引脚定义为输出，而另一些外设则改写 TRIS 位而将引脚定义为输入。用户应该参考相应的外设章节来获取更多的信息。

注： 在上电复位时，这些引脚被配置为数字输入。

外设对 TRIS 的改写会影响 TRISC 寄存器的内容。尽管外设器件可能会改写一个或多个引脚，读 TRISC 总是会返回当前内容。

例 9-4: 初始化 PORTC

```
CLRF   PORTC   ; Initialize PORTC by
             ; clearing output
             ; data latches
CLRF   LATC     ; Alternate method
             ; to clear output
             ; data latches
MOVLW  0CFh    ; Value used to
             ; initialize data
             ; direction
MOVWF  TRISC    ; Set RC<3:0> as inputs
             ; RC<5:4> as outputs
             ; RC<7:6> as inputs
```

表 9-7: PORTC I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RC0/T1OSO/ T1CKI	RC0	0	O	DIG	LATC<0> 数据输出。
		1	I	ST	PORTC<0> 数据输入。
	T1OSO	x	O	ANA	Timer1 振荡器输出；当 Timer1 振荡器使能时被使能。禁止数字 I/O。
	T1CKI	1	I	ST	Timer1 计数器输入。
RC1/T1OSI/CCP2	RC1	0	O	DIG	LATC<1> 数据输出。
		1	I	ST	PORTC<1> 数据输入。
	T1OSI	x	I	ANA	Timer1 振荡器输入；当 Timer1 振荡器使能时被使能。禁止数字 I/O。
	CCP2 ⁽¹⁾	0	O	DIG	CCP2 比较和 PWM 输出；优先级高于端口数据。
1		I	ST	CCP2 捕捉输入。	
RC2/CCP1/P1A	RC2	0	O	DIG	LATC<2> 数据输出。
		1	I	ST	PORTC<2> 数据输入。
	CCP1	0	O	DIG	ECCP1/CCP1 比较或 PWM 输出；优先级高于端口数据。
		1	I	ST	ECCP1/CCP1 捕捉输入。
P1A ⁽²⁾	0	O	DIG	ECCP1 增强型 PWM 输出，通道 A。在增强型 PWM 关闭事件期间可被配置为三态。优先级高于端口数据。	
RC3/SCK1/SCL1	RC3	0	O	DIG	LATC<3> 数据输出。
		1	I	ST	PORTC<3> 数据输入。
	SCK1	0	O	DIG	SPI 时钟输出（MSSP1 模块）；优先级高于端口数据。
		1	I	ST	SPI 时钟输入（MSSP1 模块）。
	SCL1	0	O	DIG	I ² C™ 时钟输出（MSSP1 模块）；优先级高于端口数据。
		1	I	I ² C/SMB	I ² C 时钟输入（MSSP1 模块）；输入类型取决于模块设置。
RC4/SDI1/SDA1	RC4	0	O	DIG	LATC<4> 数据输出。
		1	I	ST	PORTC<4> 数据输入。
	SDI1	1	I	ST	SPI 数据输入（MSSP1 模块）。
	SDA1	1	O	DIG	I ² C 数据输出（MSSP1 模块）；优先级高于端口数据。
		1	I	I ² C/SMB	I ² C 数据输入（MSSP1 模块）；输入类型取决于模块设置。
RC5/SDO1	RC5	0	O	DIG	LATC<5> 数据输出。
		1	I	ST	PORTC<5> 数据输入。
	SDO1	0	O	DIG	SPI 数据输出（MSSP1 模块）；优先级高于端口数据。
RC6/TX/CK	RC6	0	O	DIG	LATC<6> 数据输出。
		1	I	ST	PORTC<6> 数据输入。
	TX	1	O	DIG	异步串行发送数据输出（EUSART 模块）；优先级高于端口数据。用户必须将其配置为输出。
		1	I	ST	同步串行时钟输入（EUSART 模块）。
RC7/RX/DT	RC7	0	O	DIG	LATC<7> 数据输出。
		1	I	ST	PORTC<7> 数据输入。
	RX	1	I	ST	异步串行接收数据输入（EUSART 模块）。
	DT	1	O	DIG	同步串行数据输出（EUSART 模块）；优先级高于端口数据。
		1	I	ST	同步串行数据输入（EUSART 模块）。用户必须将其配置为输入。

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；I²C/SMB = I²C/SM 总线输入缓冲器；x = 任意值（TRIS 位不影响端口方向或者在此选项中被忽略）。

注 1: 当 CCP2MX 配置位为 0 时，对 CCP2 进行其他分配。默认配置为 RC1。

注 2: 仅在 PIC18F44J10/45J10 器件上才有增强型 PWM 输出。

PIC18F45J10 系列

表 9-8: 与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	46
LATC	PORTC 数据锁存器 (读取和写入数据锁存器)								46
TRISC	PORTC 数据方向控制寄存器								46

9.5 PORTD、TRISD 和 LATD 寄存器

注： 仅 40/44 引脚器件上有 PORTD。

PORTD 是 8 位宽的双向端口。它对应的数据方向寄存器是 TRISD。置位 TRISD (= 1) 可以让相应 PORTD 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。清零 TRISD 位 (= 0) 将使相应的 PORTD 引脚作为输出引脚（即将输出锁存器的内容置于所选择的引脚上）。

数据锁存器（LATD）也是映射的存储器。对 LATD 寄存器执行读—修改—写操作实际上是读写 PORTD 的输出锁存值。

PORTD 上的所有引脚都是用施密特触发输入缓冲器实现的。每个引脚都可被单独地设置为输入或输出。

三个 PORTD 引脚与增强型 CCP 模块的输出 P1B、P1C 和 P1D 复用。在第 14.0 节“增强型捕捉 / 比较 / PWM (ECCP) 模块”中对这些额外的 PWM 输出引脚的操作进行了详细的论述。

注： 在上电复位时，这些引脚被配置为数字输入。

通过将控制位 PSPMODE (TRISE<4>) 置 1, 可将 PORTD 配置为 8 位宽的微处理器端口（并行从动端口）。在这种模式下，输入缓冲器是 TTL 型。如需了解有关并行从动端口 (PSP) 的更多信息，请参见第 9.7 节“并行从动端口 (PSP)”。

注： 当增强型 PWM 模式使用双输出或四输出时，PORTD 的 PSP 功能被自动禁止。

例 9-5: 初始化 PORTD

```
CLRF   PORTD   ; Initialize PORTD by
               ; clearing output
               ; data latches
CLRF   LATD    ; Alternate method
               ; to clear output
               ; data latches
MOVLW  0CFh   ; Value used to
               ; initialize data
               ; direction
MOVWF  TRISD   ; Set RD<3:0> as inputs
               ; RD<5:4> as outputs
               ; RD<7:6> as inputs
```

PIC18F45J10 系列

表 9-9: PORTD I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RD0/PSP0/SCK2/SCL2	RD0	0	O	DIG	LATD<0> 数据输出。
		1	I	ST	PORTD<0> 数据输入。
	PSP0	x	O	DIG	PSP 读数据输出 (LATD<0>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
	SCK2	0	O	DIG	SPI 时钟输出 (MSSP2 模块)；优先级高于端口数据。
		1	I	ST	SPI 时钟输入 (MSSP2 模块)。
SCL2	0	O	DIG	I ² C™ 时钟输出 (MSSP2 模块)；优先级高于端口数据。	
	1	I	I ² C/SMB	I ² C 时钟输入 (MSSP2 模块)；输入类型取决于模块设置。	
RD1/PSP1/SDI2/SDA2	RD1	0	O	DIG	LATD<1> 数据输出。
		1	I	ST	PORTD<1> 数据输入。
	PSP1	x	O	DIG	PSP 读数据输出 (LATD<1>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
	SDI2	1	I	ST	SPI 数据输入 (MSSP2 模块)。
	SDA2	1	O	DIG	I ² C 数据输出 (MSSP2 模块)；优先级高于端口数据。
1		I	I ² C/SMB	I ² C 数据输入 (MSSP2 模块)；输入类型取决于模块设置。	
RD2/PSP2/SDO2	RD2	0	O	DIG	LATD<2> 数据输出。
		1	I	ST	PORTD<2> 数据输入。
	PSP2	x	O	DIG	PSP 读数据输出 (LATD<2>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
SDO2	0	O	DIG	SPI 数据输出 (MSSP2 模块)；优先级高于端口数据。	
RD3/PSP3/SS2	RD3	0	O	DIG	LATD<3> 数据输出。
		1	I	ST	PORTD<3> 数据输入。
	PSP3	x	O	DIG	PSP 读数据输出 (LATD<3>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
SS2	1	I	TTL	MSSP2 的从动选择输入 (MSSP2 模块)。	
RD4/PSP4	RD4	0	O	DIG	LATD<4> 数据输出。
		1	I	ST	PORTD<4> 数据输入。
	PSP4	x	O	DIG	PSP 读数据输出 (LATD<4>)；优先级高于端口数据。
x		I	TTL	PSP 写数据输入。	
RD5/PSP5/P1B	RD5	0	O	DIG	LATD<5> 数据输出。
		1	I	ST	PORTD<5> 数据输入。
	PSP5	x	O	DIG	PSP 读数据输出 (LATD<5>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
P1B	0	O	DIG	ECCP1 增强型 PWM 输出, 通道 B；优先级高于端口数据和 PSP 数据。在增强型 PWM 关闭事件期间可以被配置为三态。	
RD6/PSP6/P1C	RD6	0	O	DIG	LATD<6> 数据输出。
		1	I	ST	PORTD<6> 数据输入。
	PSP6	x	O	DIG	PSP 读数据输出 (LATD<6>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
P1C	0	O	DIG	ECCP1 增强型 PWM 输出, 通道 C；优先级高于端口数据和 PSP 数据。在增强型 PWM 关闭事件期间可以被配置为三态。	
RD7/PSP7/P1D	RD7	0	O	DIG	LATD<7> 数据输出。
		1	I	ST	PORTD<7> 数据输入。
	PSP7	x	O	DIG	PSP 读数据输出 (LATD<7>)；优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
P1D	0	O	DIG	ECCP1 增强型 PWM 输出, 通道 D；优先级高于端口数据和 PSP 数据。在增强型 PWM 关闭事件期间可以被配置为三态。	

图注: DIG = 数字电平输出; TTL = TTL 输入缓冲器; ST = 施密特触发器输入缓冲器; I²C/SMB = I²C/SM 总线输入缓冲器。
x = 任意值 (TRIS 位不影响端口方向或者在此选项中被忽略)。

表 9-10: 与 PORTD 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	46
LATD ⁽¹⁾	PORTD 数据锁存器 (读取和写入数据锁存器)								46
TRISD ⁽¹⁾	PORTD 数据方向控制寄存器								46
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	46
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	45

图注: — = 未用 (读为 0)。PORTD 未使用阴影单元。

注 1: 28 引脚器件上没有这些寄存器。

PIC18F45J10 系列

9.6 PORTE、TRISE 和 LATE 寄存器

注： 仅 40/44 引脚器件上有 PORTE。

在 PIC18F45J10 系列器件中，PORTE 有两种实现方式。

对于 40/44 引脚器件，PORTE 是 4 位宽端口。三个引脚（RE0/RD/AN5、RE1/WR/AN6 和 RE2/CS/AN7）可被单独地配置为输入或输出。这些引脚有施密特触发输入缓冲器。当被用作模拟输入时，这些引脚将读为 0。

PORTE 对应的数据方向寄存器是 TRISE。置位 TRISE (= 1) 可以让相应 PORTE 引脚作为输入引脚（即将相应的输出驱动器置于高阻态模式）。清零 TRISE 位 (= 0) 将使相应的 PORTE 引脚作为输出引脚（即将输出锁存器的内容置于所选择的引脚上）。

即使在 RE 引脚被用作模拟输入的时候，TRISE 寄存器仍然控制 RE 引脚的方向。此时，用户必须确保引脚的方向被配置为输入。

注： 在上电复位时，RE2:RE0 被配置为模拟输入。

TRISE 寄存器的高 4 位还控制并行从动端口的操作。在寄存器 9-1 中对这种操作进行了说明。

数据锁存器（LATE）也是映射的存储器。对 LATE 寄存器执行读—修改—写操作实际上是读写 PORTE 的输出锁存值。

例 9-6: 初始化 PORTE

```
CLRF   PORTE   ; Initialize PORTE by
              ; clearing output
              ; data latches
CLRF   LATE    ; Alternate method
              ; to clear output
              ; data latches
MOVLW  0Ah    ; Configure A/D
MOVWF  ADCON1 ; for digital inputs
MOVLW  03h    ; Value used to
              ; initialize data
              ; direction
MOVWF  TRISE  ; Set RE<0> as inputs
              ; RE<1> as outputs
              ; RE<2> as inputs
```

寄存器 9-1:

TRISE 寄存器 (仅 40/44 引脚器件)

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
bit 7							bit 0

- bit 7 **IBF:** 输入缓冲器满状态位
 1 = 已接收到一个数据, 等待 CPU 读取
 0 = 未接收到任何数据
- bit 6 **OBF:** 输出缓冲器满状态位
 1 = 输出缓冲器仍保存着上一次写入的数据
 0 = 输出缓冲器已被读取
- bit 5 **IBOV:** 输入缓冲器溢出检测位 (在微处理器模式下)
 1 = 在尚未读取上一次输入数据时又发生了一次写入 (必须由软件清零)
 0 = 没有发生溢出
- bit 4 **PSPMODE:** 并行从动端口模式选择位
 1 = 并行从动端口模式。
 0 = 通用 I/O 模式
- bit 3 **未用:** 读为 0
- bit 2 **TRISE2:** RE2 方向控制位
 1 = 输入
 0 = 输出
- bit 1 **TRISE1:** RE1 方向控制位
 1 = 输入
 0 = 输出
- bit 0 **TRISE0:** RE0 方向控制位
 1 = 输入
 0 = 输出

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

表 9-11: PORTE I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RE0/ $\overline{\text{RD}}$ /AN5	RE0	0	O	DIG	LATE<0> 数据输出；不受模拟输入影响。
		1	I	ST	PORTE<0> 数据输入；使能模拟输入时被禁止。
	$\overline{\text{RD}}$	1	I	TTL	PSP 读使能输入（PSP 使能）。
	AN5	1	I	ANA	A/D 输入通道 5；发生上电复位时的默认输入配置。
RE1/ $\overline{\text{WR}}$ /AN6	RE1	0	O	DIG	LATE<1> 数据输出；不受模拟输入影响。
		1	I	ST	PORTE<1> 数据输入；使能模拟输入时被禁止。
	$\overline{\text{WR}}$	1	I	TTL	PSP 写使能输入（PSP 使能）。
	AN6	1	I	ANA	A/D 输入通道 6；发生上电复位时的默认输入配置。
RE2/ $\overline{\text{CS}}$ /AN7	RE2	0	O	DIG	LATE<2> 数据输出；不受模拟输入影响。
		1	I	ST	PORTE<2> 数据输入；使能模拟输入时被禁止。
	$\overline{\text{CS}}$	1	I	TTL	PSP 写使能输入（PSP 使能）。
	AN7	1	I	ANA	A/D 输入通道 7；发生上电复位时的默认输入配置。

图注： DIG = 数字电平输出； TTL = TTL 输入缓冲器； ST = 施密特触发器输入缓冲器； ANA = 模拟电平输入 / 输出； x = 任意值（TRIS 位不影响端口方向或者在此选项中被忽略）。

表 9-12: 与 PORTE 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
PORTE ⁽¹⁾	—	—	—	—	—	RE2	RE1	RE0	46
LATE ⁽¹⁾	—	—	—	—	—	PORTE 数据锁存器 (读取和写入数据锁存器)			46
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	46
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	44

图注： — = 未用位（读为 0）。PORTE 未使用阴影单元。

注 1： 28 引脚器件上没有这些寄存器。

9.7 并行从动端口 (PSP)

注: 仅 40/44 引脚器件上有并行从动端口。

除了充当通用 I/O 端口之外, PORTD 还可以作为 8 位并行从动端口 (PSP) 或微处理器端口工作。PSP 的工作由 TRISE 寄存器 (寄存器 9-1) 的高 4 位控制。只要增强型 CCP 模块没有工作在双输出或四输出 PWM 模式下, 就可以通过置位控制位 PSMODE (TRISE<4>) 使能 PSP 操作。在从动模式下, 外设可异步读写该端口。

PSP 可以直接与 8 位微处理器数据总线进行接口。外部微处理器可以将 PORTD 锁存器作为一个 8 位锁存器进行读或写。将 PSMODE 控制位置 1, 使 PORTE I/O 引脚成为微处理器端口的控制输入端。置 1 后, 端口引脚 RE0 成为 RD 输入, RE1 成为 WR 输入, 而 RE2 成为 CS (片选) 输入。用作此功能的时候, 必须将 TRISE 寄存器 (TRISE<2:0>) 中对应的数据方向位配置为输入 (置 1)。A/D 端口配置位 PFCG3:PFCG0 (ADCON1<3:0>) 也必须被设置为 1010 到 1111 范围内的一个值。

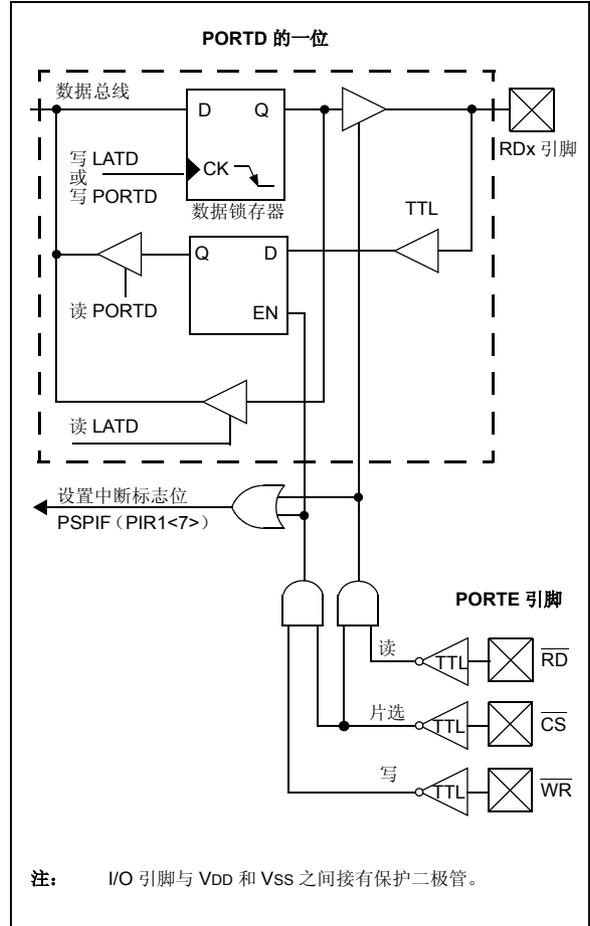
当第一次检测到 \overline{CS} 和 \overline{WR} 线为低电平时, 开始写 PSP, 当检测到它们两者之一为高电平时, 写 PSP 结束。当写操作结束时, PSPIF 和 IBF 标志位都被置 1。

当第一次检测到 \overline{CS} 和 \overline{RD} 线为低电平时, 开始读 PSP。此时 PORTD 中的数据被读出且 OBF 位被清零。如果用户将新数据写入 PORTD 以试图置位 OBF, 该数据将被立即读出, 但 OBF 位不会被置 1。

当检测到 \overline{CS} 或 \overline{RD} 线为高电平时, PORTD 引脚返回到输入状态并且 PSPIF 位被置 1。用户应用程序在为 PSP 提供服务之前应该先等待 PSPIF 被置 1; 当 PSPIF 被置 1 后, 可以查询 IBF 和 OBF 位并进行相应的操作。

图 9-4 和图 9-5 分别给出了在读和写两种模式下的控制信号的时序。

图 9-3: PORTD 和 PORTE 框图 (并行从动端口)



PIC18F45J10 系列

图 9-4: 并行从动端口写操作时序

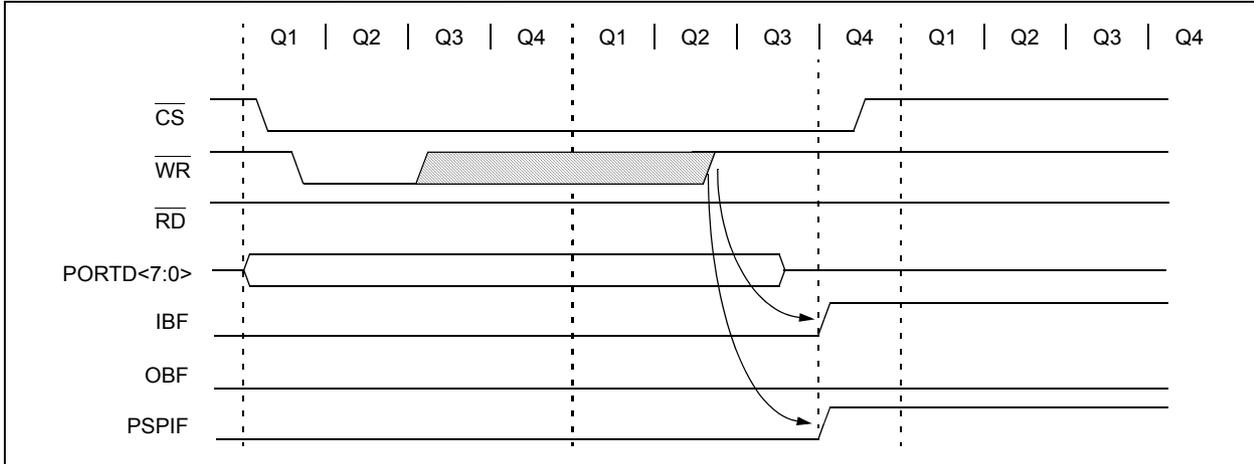


图 9-5: 并行从动端口读操作时序

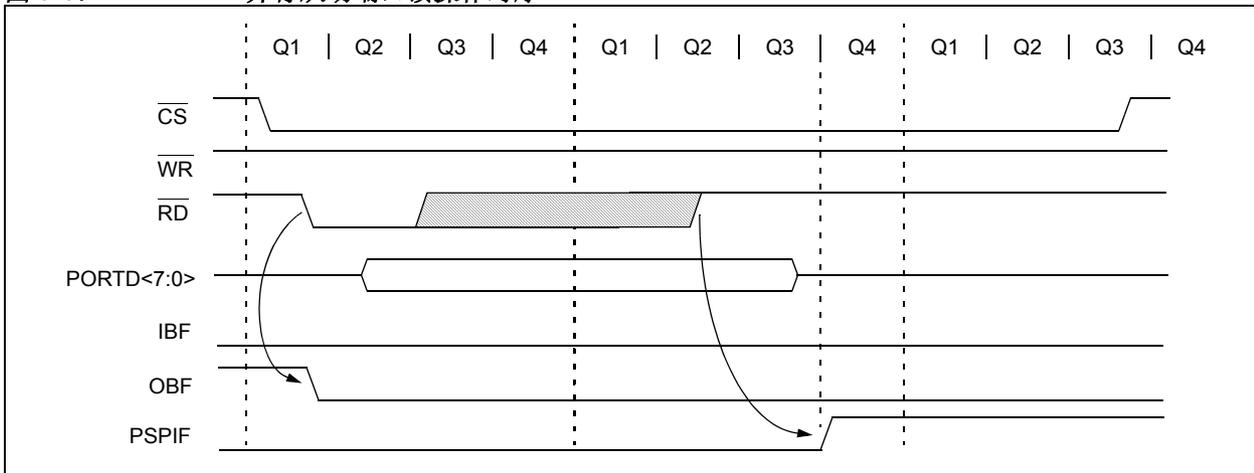


表 9-13: 与并行从动端口相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	46
LATD ⁽¹⁾	PORTD 数据锁存器 (读取和写入数据锁存器)								46
TRISD ⁽¹⁾	PORTD 数据方向控制寄存器								46
PORTE ⁽¹⁾	—	—	—	—	—	RE2	RE1	RE0	46
LATE ⁽¹⁾	—	—	—	—	—	PORTE 数据锁存器 (读取和写入数据锁存器)			46
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	46
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	44

图注: — = 未用位 (读为 0)。并行从动端口不使用阴影单元。

注 1: 这些寄存器和 / 或位在 28 引脚器件上不存在, 应该被读为 0。

10.0 TIMER0 模块

Timer0 模块具有以下功能:

- 可通过软件选择, 作为 8 位或 16 位定时器 / 计数器
- 可读写的寄存器
- 专用的 8 位软件可编程预分频器
- 可选的时钟源 (内部或外部)
- 外部时钟的边沿选择
- 溢出中断

T0CON 寄存器 (寄存器 10-1) 控制模块工作的所有方面, 包括预分频比选择。该寄存器是可读写的。

图 10-1 显示了 8 位模式下 Timer0 模块的简化框图, 图 10-2 显示了 16 位模式下 Timer0 模块的简化框图。

寄存器 10-1: T0CON: TIMER0 控制寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

- bit 7 **TMR0ON:** Timer0 开 / 关控制位
 1 = 使能 Timer0
 0 = 停止 Timer0
- bit 6 **T08BIT:** Timer0 8 位 / 16 位控制位
 1 = Timer0 被配置为 8 位定时器 / 计数器
 0 = Timer0 被配置为 16 位定时器 / 计数器
- bit 5 **T0CS:** Timer0 时钟源选择位
 1 = T0CKI 引脚上的传输信号作为时钟
 0 = 使用内部指令周期时钟 (CLKO) 作为时钟
- bit 4 **T0SE:** Timer0 时钟源边沿选择位
 1 = 在 T0CKI 引脚上信号的下降沿触发递增
 0 = 在 T0CKI 引脚上信号的上升沿触发递增
- bit 3 **PSA:** Timer0 预分频器分配位
 1 = 未分配给 Timer0 预分频器。Timer0 时钟输入不经过预分频器。
 0 = 已分配给 Timer0 预分频器。Timer0 时钟输入信号来自预分频器的输出。
- bit 2-0 **T0PS2:T0PS0:** Timer0 预分频比选择位
 111 = 1:256 预分频比
 110 = 1:128 预分频比
 101 = 1:64 预分频比
 100 = 1:32 预分频比
 011 = 1:16 预分频比
 010 = 1:8 预分频比
 001 = 1:4 预分频比
 000 = 1:2 预分频比

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

10.1 Timer0 工作原理

Timer0 既可用作定时器亦可用作计数器；具体的模式由 TOCS 位 (T0CON<5>) 选择。在定时器模式下 (TOCS = 0)，除非选择了不同的预分频比，否则，默认情况下在每个时钟周期 Timer0 模块都会加 1 (见第 10.3 节“预分频器”)。如果写入 TMR0 寄存器，那么在随后的两个指令周期，它将不再加 1。用户可通过将校正值写入 TMR0 寄存器解决此问题。

通过将 TOCS 位置 1 选择计数器模式。在计数器模式下，Timer0 可在 RB5/T0CKI 引脚信号的每个上升沿或下降沿递增计数。触发递增的边沿由 Timer0 时钟源边沿选择位 T0SE (T0CON<4>) 决定。清零此位选择上升沿触发。下面讨论外部时钟输入的限制条件。

可以使用外部时钟源来驱动 Timer0，但是必须满足一定的要求以确保外部时钟与内部时钟 (Tosc) 相位同步。在同步之后，定时器 / 计数器仍需要一定的延时才会引发递增操作。

10.2 Timer0 的 16 位读写模式

在 16 位模式下，TMR0H 并非 Timer0 的高字节。它实际上是被缓存的 Timer0 的高字节，Timer0 的高字节是不可以直接读写的 (见图 10-2)。在读 TMR0L 时使用 Timer0 高字节的内容更新 TMR0H。这样可以一次读取 Timer0 的全部 16 位，而无需验证读到的高字节和低字节的有效性。(在高、低字节分两次连续读取的情况下，由于可能存在进位，所以需要验证读到字节的有效性)。

同样，可以使用 TMR0H 缓冲寄存器写入 Timer0 的高字节。在写入 TMR0L 的同时，使用 TMR0H 的内容更新 Timer0 的高字节。这样一次就可以完成 Timer0 全部 16 位的更新。

图 10-1: TIMER0 框图 (8 位模式)

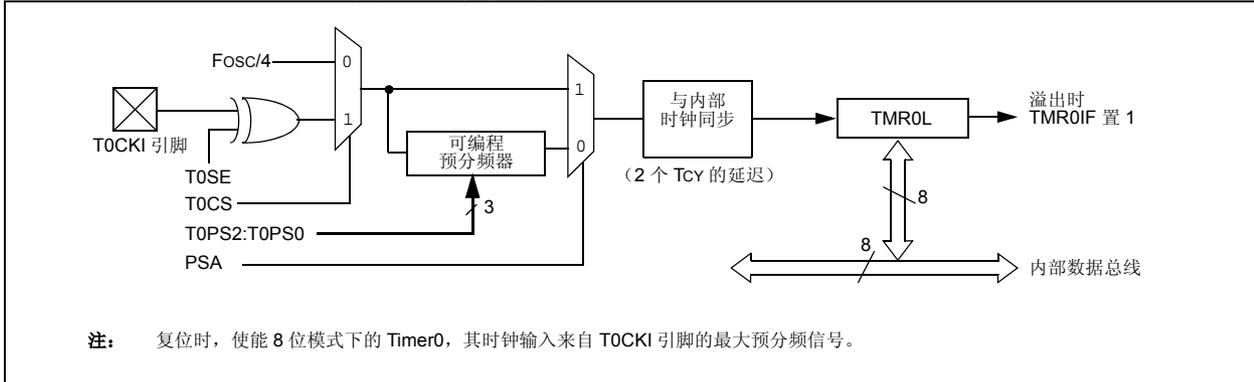
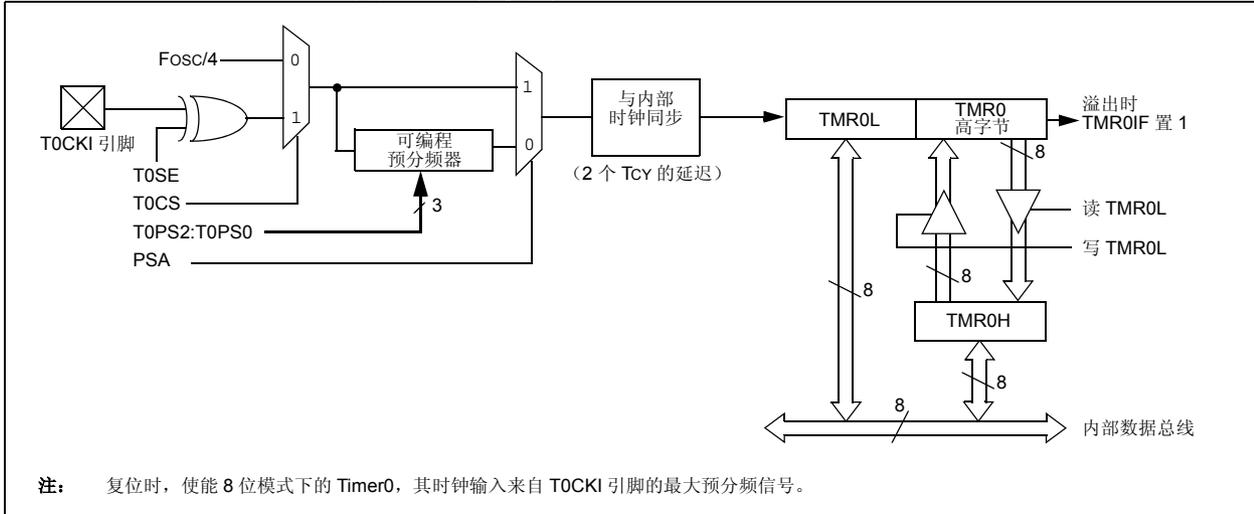


图 10-2: TIMER0 框图 (16 位模式)



10.3 预分频器

Timer0 模块的预分频器为一个 8 位计数器。该预分频器不可被直接读写。设置 PSA 和 T0PS2:T0PS0 位 (T0CON<3:0>) 可分别决定预分频器的分配和预分频比。

把 PSA 位清零可将预分频器分配给 Timer0 模块。在分配预分频器时, 预分频比可以在 1:2 到 1:256 之间进行选择, 该分频比以 2 的整数幂递增。

当将预分频器分配给 Timer0 模块时, 所有写入 TMR0 寄存器的指令 (如 CLRF TMR0、MOVWF TMR0 和 BSF TMR0 等) 都将使预分频器的计数值清零。

注: 当将预分频器分配给 Timer0 时, 写入 TMR0 会将预分频器的计数值清零, 但不会改变预分频器的分配。

10.3.1 切换预分频器

预分频器的分配完全由软件控制, 并且在程序执行期间可以随时更改。

10.4 Timer0 中断

当 TMR0 寄存器发生溢出时 (8 位模式下, 从 FFh 到 00h; 或 16 位模式下, 从 FFFFh 到 0000h), 将产生 TMR0 中断。这种溢出会将标志位 TMR0IF 置 1。清零 TMR0IE 位 (INTCON<5>) 可屏蔽此中断。在重新允许该中断前, 必须在中断服务程序中用软件清零 TMR0IF 位。

由于 Timer0 在休眠模式下是关闭的, 所以 TMR0 中断无法将处理器从休眠状态唤醒。

表 10-1: 与 TIMER0 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TMR0L	Timer0 寄存器的低字节								44
TMR0H	Timer0 寄存器的高字节								44
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	44
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46

图注: — = 未用 (读为 0)。Timer0 不使用阴影单元。

PIC18F45J10 系列

注:

11.0 TIMER1 模块

Timer1 定时器 / 计数器模块具有以下功能:

- 可通过软件选择, 作为 16 位定时器或计数器
- 可读写的 8 位寄存器 (TMR1H 和 TMR1L)
- 可选择使用器件时钟或 Timer1 内部振荡器作为时钟源
- 溢出中断
- CCP 特殊事件触发复位
- 器件时钟状态标志位 (T1RUN)

图 11-1 是 Timer1 模块的简化框图。图 11-2 是此模块在读 / 写模式下工作的框图。

该模块自身具有低功耗振荡器, 可提供额外的时钟。Timer1 振荡器也可作为单片机处于功耗管理模式下的时钟源。

在对外部元件数量和代码开销要求苛刻的应用中, 可使用 Timer1 为其提供实时时钟 (Real-Time Clock, RTC)。

Timer1 的工作由 T1CON 控制寄存器 (寄存器 11-1) 控制。该寄存器包含 Timer1 振荡器使能位 (T1OSCEN)。可以通过将控制位 TMR1ON (T1CON<0>) 置 1 或清零来使能或禁止 Timer1。

寄存器 11-1: T1CON: TIMER1 控制寄存器

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7						bit 0	

- bit 7 **RD16:** 16 位读 / 写模式使能位
 1 = 使能通过一次 16 位操作对 Timer1 寄存器进行读写
 0 = 使能通过两次 8 位操作对 Timer1 寄存器进行读写
- bit 6 **T1RUN:** Timer1 系统时钟状态位
 1 = 器件时钟由 Timer1 振荡器产生
 0 = 器件时钟由另一个时钟源产生
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 输入时钟预分频比选择位
 11 = 1:8 预分频比
 10 = 1:4 预分频比
 01 = 1:2 预分频比
 00 = 1:1 预分频比
- bit 3 **T1OSCEN:** Timer1 振荡器使能位
 1 = 使能 Timer1 振荡器
 0 = 关闭 Timer1 振荡器
 关闭振荡器的反相器和反馈电阻以降低功耗。
- bit 2 **T1SYNC:** Timer1 外部时钟输入同步选择位
当 TMR1CS = 1 时:
 1 = 不与外部时钟输入同步
 0 = 与外部时钟输入同步
当 TMR1CS = 0 时:
 忽略此位。当 TMR1CS=0 时, Timer1 使用内部时钟。
- bit 1 **TMR1CS:** Timer1 时钟源选择位
 1 = 使用 RC0/T1OSO/T1CKI 引脚上的外部时钟 (上升沿触发计数)
 0 = 使用内部时钟 (Fosc/4)
- bit 0 **TMR1ON:** Timer1 使能位
 1 = 使能 Timer1
 0 = 禁止 Timer1

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

11.1 Timer1 工作原理

Timer1 可工作在以下模式：

- 定时器
- 同步计数器
- 异步计数器

工作模式由时钟选择位 TMR1CS (T1CON<1>) 决定。当 TMR1CS 清零 (= 0) 时，Timer1 在每个内部指令

周期 ($F_{osc}/4$) 递增。当 TMR1CS 位置 1 时，Timer1 在 Timer1 外部时钟输入信号或 Timer1 振荡器 (如果能) 输出信号的每个上升沿递增。

当使能 Timer1 时，RC1/T1OSI 和 RC0/T1OSO/T1CKI 引脚变为输入引脚。这意味着 TRISC<1:0> 的值被忽略并且这些引脚的读取值为 0。

图 11-1: TIMER1 框图

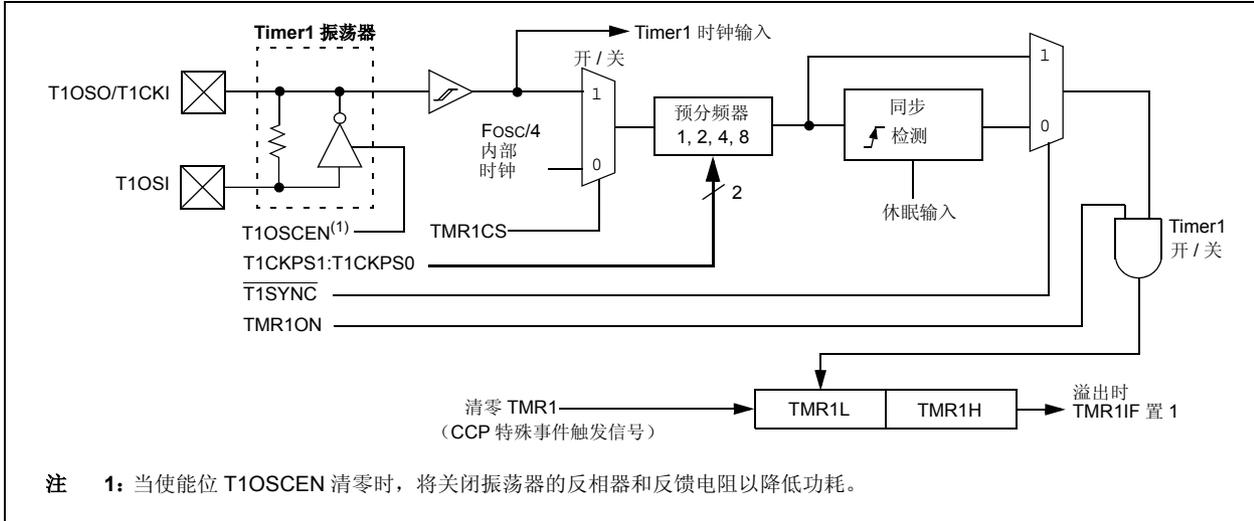
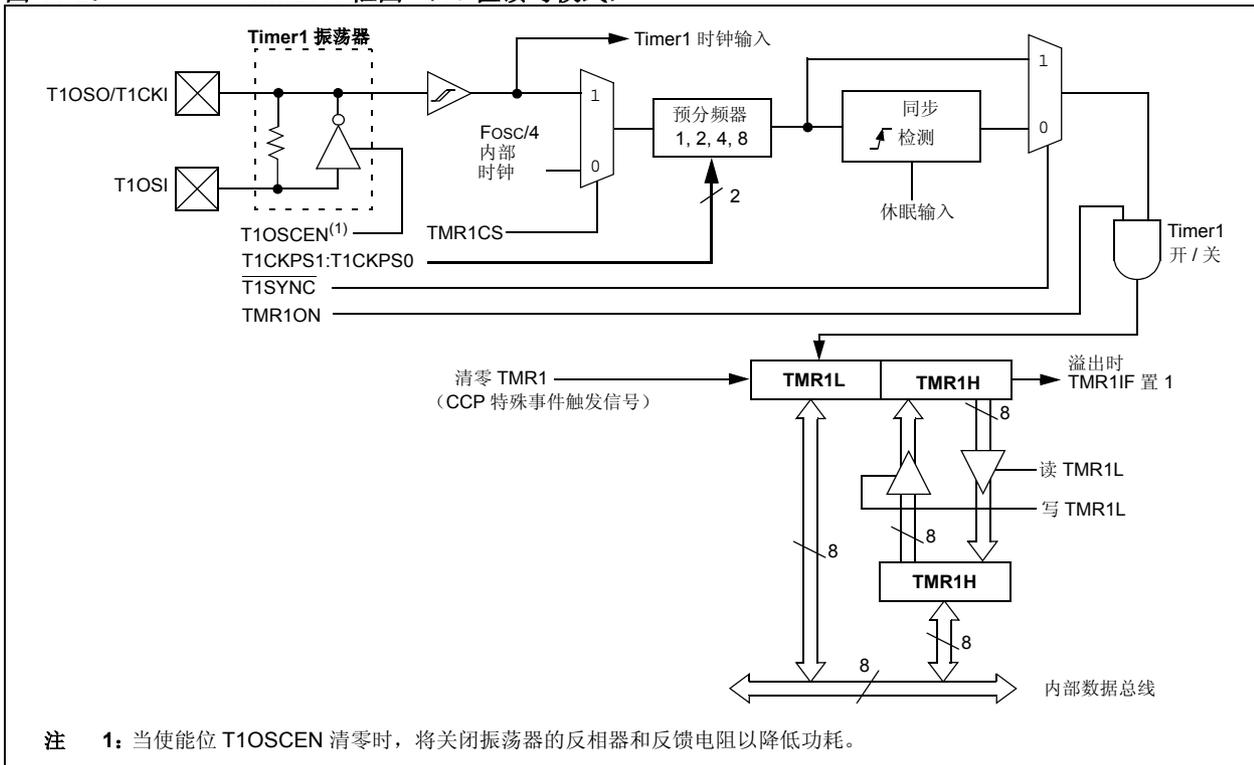


图 11-2: TIMER1 框图 (16 位读写模式)



11.2 Timer1 16 位读 / 写模式

可将 Timer1 配置为 16 位读写模式（见图 11-2）。当 RD16 控制位（T1CON<7>）置 1 时，TMR1H 的地址被映射到 Timer1 的高字节缓冲寄存器。对 TMR1L 的读操作将把 Timer1 的高字节内容装入 Timer1 高字节缓冲寄存器。这种方式使用户可以精确地读取 Timer1 的全部 16 位，而不需要像先读高字节再读低字节那样由于两次读取之间可能存在进位，而不得不验证读取的有效性。

对 Timer1 的高字节进行写操作也必须通过 TMR1H 缓冲器进行。在写入 TMR1L 的同时，使用 TMR1H 的内容更新 Timer1 的高字节。这样允许用户将 16 位值一次写入 Timer1 的高字节和低字节。

在该模式下不能直接读写 Timer1 的高字节。所有读写都必须通过 Timer1 高字节缓冲器进行。写入 TMR1H 不会清零 Timer1 预分频器。只有在写 TMR1L 时才会清零预分频器。

11.3 Timer1 振荡器

片内晶体振荡器电路连接在 T1OSI（输入）引脚和 T1OSO（放大器输出）引脚之间。通过将 Timer1 振荡器使能位 T1OSCEN（T1CON<3>）置 1 可启用该振荡器电路。此振荡器电路是一种低功耗电路，它采用了额定振荡频率为 32 kHz 的晶振，在所有的功耗管理模式下都可继续运行。图 11-3 所示为典型的振荡器电路。表 11-1 显示了 Timer1 振荡器的电容选择。

用户必须提供软件延迟来确保 Timer1 振荡器的正常起振。

表 11-1: Timer1 振荡器的电容选择^(2,4)

振荡器类型	频率	C1	C2
LP	32 kHz	27 pF ⁽¹⁾	27 pF ⁽¹⁾

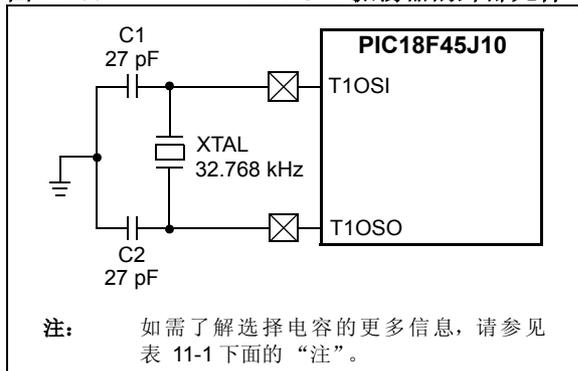
- 注
- 1: Microchip 建议在验证振荡电路时这些值仅供参考。
 - 2: 选用较大的电容值虽然可以提高振荡器的稳定性，但同时也会延长起振时间。
 - 3: 由于谐振器 / 晶振的特性各不相同，因此用户应当向谐振器 / 晶振制造厂商咨询外部元件的相应值。
 - 4: 上述电容值仅供设计参考。

11.3.1 使用 Timer1 作为时钟源

在功耗管理模式中也可以将 Timer1 振荡器用作时钟源。通过将时钟选择位 SCS1:SCS0（OSCCON<1:0>）设置为 01，器件可以切换到 SEC_RUN 模式，此时 CPU 和外设都可以用 Timer1 振荡器作为时钟源。如果 IDLEN 位（OSCCON<7>）被清零并且执行了 SLEEP 指令，器件将进入 SEC_IDLE 模式。欲知更多详情，请参见第 3.0 节“功耗管理模式”。

只要 Timer1 振荡器提供器件时钟，Timer1 系统时钟状态标志位 T1RUN（T1CON<6>）就会置 1。这可用于确定控制器的当前时钟模式。该位也可指示故障保护时钟监视器当前正使用的时钟源。如果使能了故障保护时钟监视器并且 Timer1 振荡器在提供时钟信号时发生了故障，轮询 T1RUN 位可以确定时钟源是 Timer1 振荡器还是其他时钟源。

图 11-3: Timer1 振荡器的外部元件



PIC18F45J10 系列

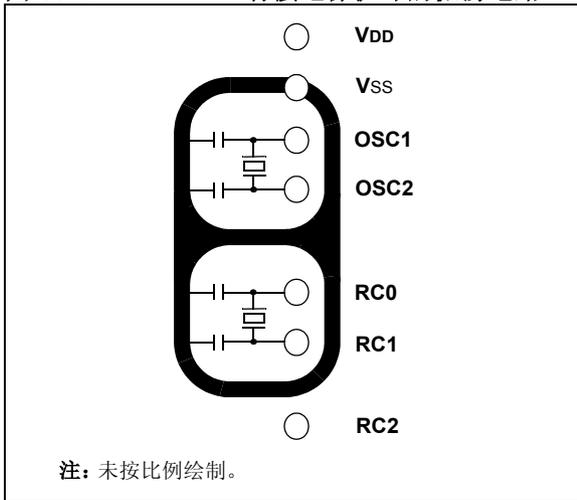
11.3.2 Timer1 振荡器布线注意事项

Timer1 振荡器在工作期间消耗极少的功率。鉴于此振荡器的低功耗特性，它对附近变化较快的信号可能会比较敏感。

如图 11-3 所示，振荡电路应该尽可能靠近单片机。除了 Vss 或 VDD 外，在振荡电路区域内不应有其他电路。

如果必须要在该振荡器附近布置高速电路（如输出比较模式或 PWM 模式下的 CCP1 引脚，或使用 OSC2 引脚的主振荡器），那么在该振荡电路周围布置接地保护环（如图 11-4 所示），对于单面 PCB 板或外加接地层的 PCB 板来讲可能会有帮助。

图 11-4: 有接地保护环的振荡电路



11.4 Timer1 中断

TMR1 寄存器对 (TMR1H:TMR1L) 从 0000h 开始递增，直到 FFFFh，然后溢出从 0000h 重新开始计数。如果使能了 Timer1 中断，该中断就会在溢出时产生，并将中断标志位 TMR1IF (PIR1<0>) 置 1。可以通过对 Timer1 中断允许位 TMR1IE (PIE1<0>) 置 1 或清零来允许或禁止该中断。

11.5 使用 ECCP/CCP 特殊事件触发信号复位 Timer1

如果 ECCP1/CCP1 或 CCP2 被配置为比较模式 (CCPxM3:CCPxM0 = 1011)，并且产生特殊事件触发信号，该信号将复位 Timer1。如果使能了 A/D 模块，来自 CCP2 的触发信号还将启动 A/D 转换（欲知更多信息，请参见第 14.2.1 节“特殊事件触发器”）。

要使用这一功能，必须将 Timer1 配置为定时器或同步计数器。在这种情况下，CCPRH:CCPRL 这对寄存器实际上变成了 Timer1 的周期寄存器。

如果 Timer1 工作在异步计数器模式下，复位操作将不起作用。

如果对 Timer1 的写操作和特殊事件触发信号同时发生，则写操作优先。

注：来自 ECCP1/CCPx 模块的特殊事件触发信号不会将中断标志位 TMR1IF (PIR1<0>) 置 1。

11.6 使用 Timer1 作为实时时钟

为 Timer1 添加外部 LP 振荡器（如第 11.3 节“Timer1 振荡器”中所述），可以为用户提供 RTC 功能。这是通过一个提供精确时基的廉价时钟晶振以及几行计算时间的应用程序代码实现的。当器件工作于休眠模式下并使用电池或超大容量电容作为电源时，可以省去额外的 RTC 器件和备用电池。

应用代码程序 RTCISR（如例 11-1 所示），给出了使用中断服务程序以 1 秒的间隔递增计数器的简单方法。将 TMR1 寄存器对的值不断加 1 直至溢出，触发中断并调用中断服务程序，该程序会使秒计数器加 1，而分钟和小时计数器则会在前面的计数器溢出时加 1。

由于这对寄存器为 16 位宽，因此使用 32.768 kHz 时钟，将其计数到溢出需要 2 秒。要使溢出按所需的 1 秒间隔进行，必须预先装载这对寄存器。最简单的方法是使用 BSF 指令将 TMR1H 的最高有效位置 1。请注意决不要预先加载或改变 TMR1L 寄存器，这样做可能会引起多个周期的累积错误。

要使此方法够精确，如程序 RTCinit 所示，Timer1 必须工作在异步模式且必须使能 Timer1 溢出中断 (PIE1<0> = 1)。同时 Timer1 振荡器也必须使能并始终保持运行。

例 11-1: 使用 TIMER1 中断服务实现实时时钟

```

RTCinit
    MOVLW    80h                ; Preload TMR1 register pair
    MOVWF   TMR1H              ; for 1 second overflow
    CLRF    TMR1L
    MOVLW   b'00001111'        ; Configure for external clock,
    MOVWF   T1CON              ; Asynchronous operation, external oscillator
    CLRF    secs               ; Initialize timekeeping registers
    CLRF    mins               ;
    MOVLW   .12
    MOVWF   hours
    BSF     PIE1, TMR1IE       ; Enable Timer1 interrupt
    RETURN

RTCisr
    BSF     TMR1H, 7           ; Preload for 1 sec overflow
    BCF     PIR1, TMR1IF      ; Clear interrupt flag
    INCF    secs, F           ; Increment seconds
    MOVLW   .59               ; 60 seconds elapsed?
    CPFSGT  secs
    RETURN                        ; No, done
    CLRF    secs              ; Clear seconds
    INCF    mins, F          ; Increment minutes
    MOVLW   .59               ; 60 minutes elapsed?
    CPFSGT  mins
    RETURN                        ; No, done
    CLRF    mins              ; clear minutes
    INCF    hours, F         ; Increment hours
    MOVLW   .23               ; 24 hours elapsed?
    CPFSGT  hours
    RETURN                        ; No, done
    CLRF    hours             ; Reset hours
    RETURN                        ; Done
    
```

表 11-2: Timer1 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
TMR1L	Timer1 寄存器的低字节								44
TMR1H	Timer1 寄存器的高字节								44
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCN	T1SYNC	TMR1CS	TMR1ON	44

图注: Timer1 模块不使用阴影单元。

注 1: 这些位在 28 引脚器件中不存在, 始终读为 0。

PIC18F45J10 系列

注:

12.0 TIMER2 模块

Timer2 定时器模块具有以下功能：

- 8 位定时器和周期寄存器（分别为 TMR2 和 PR2）
- 可读写（以上两个寄存器）
- 可软件编程的预分频器（分频比为 1:1、1:4 和 1:16）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR2 与 PR2 匹配时产生中断
- 作为 MSSP 模块的可选移位时钟

此模块的工作由 T2CON 寄存器（寄存器 12-1）控制，此寄存器使能或禁止定时器并配置预分频器和后分频器。可以通过清零控制位 TMR2ON（T2CON<2>）关闭 Timer2，以实现最小功耗。

图 12-1 所示为该模块的简化框图。

12.1 Timer2 工作原理

在正常操作中，TMR2 从 00h 开始，每个时钟周期（Fosc/4）计数一次。4 位的计数器 / 预分频器提供了对时钟输入不分频、4 分频和 16 分频三种选项，可通过预分频控制位 T2CKPS1:T2CKPS0（T2CON<1:0>）进行选择。在每个时钟周期，TMR2 的值都会与周期寄存器 PR2 中的值进行比较。当两个值匹配时，由比较器产生匹配信号作为定时器的输出。此信号也会使 TMR2 的值在下一个周期复位到 00h，并驱动输出计数器 / 后分频器（见第 12.2 节“Timer2 中断”）。

TMR2 和 PR2 寄存器均可直接读写。在器件复位时，TMR2 寄存器都会清零，而 PR2 寄存器则初始化为 FFh。发生以下事件时，预分频和后分频计数器均会清零：

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 任何方式的器件复位（上电复位、 $\overline{\text{MCLR}}$ 复位、看门狗定时器复位或者欠压复位）

写 T2CON 时 TMR2 不会清零。

寄存器 12-1: T2CON: TIMER2 控制寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 未用：读为 0

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 输出后分频比选择位

0000 = 1:1 后分频比

0001 = 1:2 后分频比

•

•

•

1111 = 1:16 后分频比

bit 2 **TMR2ON**: Timer2 使能位

1 = Timer2 打开

0 = Timer2 关闭

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 时钟预分频值选择位

00 = 预分频值为 1

01 = 预分频值为 4

1x = 预分频值为 16

图注:

R = 可读位

W = 可写位

U = 未用位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

12.2 Timer2 中断

Timer2 也可以产生可选的器件中断。Timer2 输出信号 (TMR2 和 PR2 匹配时) 可作为 4 位输出计数器 / 后分频器的输入信号。此计数器产生的 TMR2 匹配中断, 由其标志位 TMR2IF (PIR1<1>) 表示。可以通过将 TMR2 匹配中断允许位 TMR2IE (PIE1<1>) 置 1 来允许此中断。

可以通过后分频器控制位 T2OUTPS3:T2OUTPS0 (T2CON<6:3>) 在 16 个后分频比选项 (从 1:1 到 1:16) 中进行选择。

12.3 Timer2 输出

TMR2 的不经分频的输出主要用于 CCP 模块, 它用作 CCP 模块在 PWM 模式下工作时的时基。

还可将 Timer2 用作 MSSP 模块在 SPI 模式下的移位时钟源。第 15.0 节“主控同步串行端口 (MSSP) 模块”提供了更多信息。

图 12-1: Timer2 框图

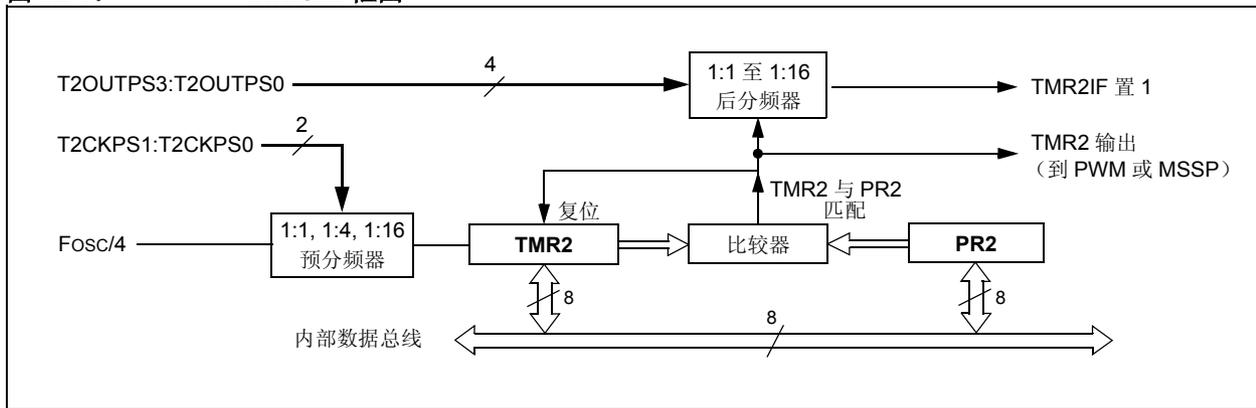


表 12-1: Timer2 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
TMR2	Timer2 寄存器								44
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	44
PR2	Timer2 周期寄存器								44

图注: — = 未用 (读为 0)。Timer2 模块不使用阴影单元。

注 1: 这些位在 28 引脚器件中不存在, 始终读为 0。

13.0 捕捉 / 比较 / PWM (CCP) 模块

PIC18F45J10 系列器件都有两个 CCP (捕捉 / 比较 / PWM) 模块。每个模块包含一个 16 位寄存器, 它可以用作 16 位捕捉寄存器、16 位比较寄存器或 PWM 主 / 从占空比寄存器。

在 28 引脚器件中, 两个标准 CCP 模块 (CCP1 和 CCP2) 以本章中描述的方式操作。在 40/44 引脚器件中, CCP1 模块是增强型 CCP 模块 (ECCP1), 具有标准的捕捉和比较模式以及增强型 PWM 模式。第 14.0 节“增强型捕捉 / 比较 / PWM (ECCP) 模块”将讨论增强型 CCP 模块的使用。

在本章中描述的捕捉和比较操作适用于所有标准和增强型 CCP 模块。

注: 在本章节和第 14.0 节“增强型捕捉 / 比较 / PWM (ECCP) 模块”中, 在提到 CCP 模块的寄存器和位名称时, 一般会使用“x”或“y”代替特定的模块编号。因此, “CCPxCON”可能指 CCP1、CCP2 或 ECCP1 的控制寄存器。相关章节中都使用“CCPxCON”来指模块控制寄存器, 不管使用的是标准还是增强型 CCP 模块。

寄存器 13-1: CCPxCON 寄存器 (28 引脚器件中的 CCP1 和 CCP2 模块)

	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7								bit 0

bit 7-6 未用: 读为 0

bit 5-4 **DCxB1:DCxB0:** PWM 占空比 bit 1 和 bit 0

捕捉模式:

未用。

比较模式:

未用。

PWM 模式:

这两位是 10 位 PWM 占空比的低 2 位 (bit 1 和 bit 0)。占空比的高 8 位 (DCx9:DCx2) 在 CCPxL 中。

bit 3-0 **CCPxM3:CCPxM0:** CCPx 模式选择位

0000 = 禁止捕捉 / 比较 / PWM (复位 CCPx 模块)

0001 = 保留

0010 = 比较模式, 匹配时输出电平翻转 (CCPxIF 位置 1)

0011 = 保留

0100 = 捕捉模式, 在每个下降沿发生捕捉

0101 = 捕捉模式, 在每个上升沿发生捕捉

0110 = 捕捉模式, 每 4 个上升沿发生捕捉

0111 = 捕捉模式, 每 16 个上升沿发生捕捉

1000 = 比较模式: CCPx 引脚初始化为低电平; 比较匹配时, 强制 CCPx 引脚为高电平 (CCPxIF 位置 1)

1001 = 比较模式: CCPx 引脚初始化为高电平; 比较匹配时, 强制 CCPx 引脚为低电平 (CCPxIF 位置 1)

1010 = 比较模式: 比较匹配时产生软件中断 (CCPxIF 位置 1,

CCPx 引脚反映出 I/O 状态)

1011 = 比较模式: CCPx 匹配时触发特殊事件, 复位定时器, 启动 A/D 转换 (CCPxIF 位置 1)

11xx = PWM 模式

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

13.1 CCP 模块配置

每个捕捉 / 比较 / PWM 模块均与一个控制寄存器（通常为 CCPxCON）和一个数据寄存器（CCPRx）相对应。数据寄存器由 2 个 8 位寄存器组成：CCPRxL（低字节）和 CCPRxH（高字节）。所有寄存器都是可读写的。

13.1.1 CCP 模块和定时器资源

CCP 模块依据所选择的工作模式，使用 Timer1 或 2。Timer1 适用于在捕捉或比较模式下的模块，而 Timer2 适用于在 PWM 模式下的模块。

表 13-1: ECCP/CCP 模式一定时器资源

ECCP/CCP 模式	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

如果将两个模块配置为同时在相同的模式（捕捉 / 比较或 PWM）下工作，那么这两个模块在任一给定时刻均处于激活状态，并可共用相同的定时器资源。图 13-1 和图 13-2 为两个模块之间的关系汇总。处于异步计数器模式的 Timer1，无法进行捕捉操作。

13.1.2 CCP2 引脚分配

根据器件配置，CCP2 的引脚分配（捕捉输入、比较和 PWM 输出）可以更改。CCP2MX 配置位决定哪个引脚将与 CCP2 复用。默认情况下，CCP2 引脚被分配给 RC1（CCP2MX = 1）。如果该配置位清零，则 CCP2 与 RB3 复用。

改变 CCP2 的引脚分配不会自动更改配置该端口引脚的寄存器内容。用户必须总是确认已为 CCP2 操作正确地配置了相应的 TRIS 寄存器，无论该寄存器的位置在哪里。

表 13-2: ECCP1/CCP1 和 CCP2 使用定时器资源的相互关系

CCP1 模式	CCP2 模式	相互关系
捕捉	捕捉	使用 TMR1 作为时基的模块。
捕捉	比较	可将 CCP2 配置为特殊事件触发器以复位 TMR1。也可以在发生触发事件时进行自动 A/D 转换。ECCP1/CCP1 的操作将受到影响。
比较	捕捉	可将 ECCP1/CCP1 模块配置为特殊事件触发器以复位 TMR1。CCP2 的操作将受到影响。
比较	比较	可将这两个模块配置为特殊事件触发器以复位 TMR1。在发生 CCP2 触发事件时可进行自动 A/D 转换。
捕捉	PWM ⁽¹⁾	无
比较	PWM ⁽¹⁾	无
PWM ⁽¹⁾	捕捉	无
PWM ⁽¹⁾	比较	无
PWM ⁽¹⁾	PWM	两个 PWM 将具有相同的频率和更新速度（TMR2 中断）。

注 1: 包括标准和增强型 PWM 操作。

13.2 捕捉模式

在捕捉模式下，当对应的 CCPx 引脚发生以下事件时，CCPRxH:CCPRxL 寄存器对捕捉 TMR1 寄存器的 16 位值。事件定义如下：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

通过模式选择位 CCPxM3:CCPxM0 (CCPxCON<3:0>) 选择事件类型。当一个捕捉发生时，中断请求标志位 CCPxIF 置 1；它必须用软件清零。如果在寄存器 CCPRx 中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

13.2.1 CCP 引脚配置

在捕捉模式下，应该通过将对应的 TRIS 方向位置 1 来将相应的 CCPx 引脚配置为输入。

注： 如果 RB3/CCP2 或 RC1/CCP2 引脚被配置为输出，对该端口的写操作可能引发一个捕捉事件。

13.2.2 软件中断

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该保持 CCPxIE 中断允许位清零以避免错误中断。在操作模式发生任何改变之后也应该清零中断标志位 CCPxIF。

13.2.3 CCP 预分频器

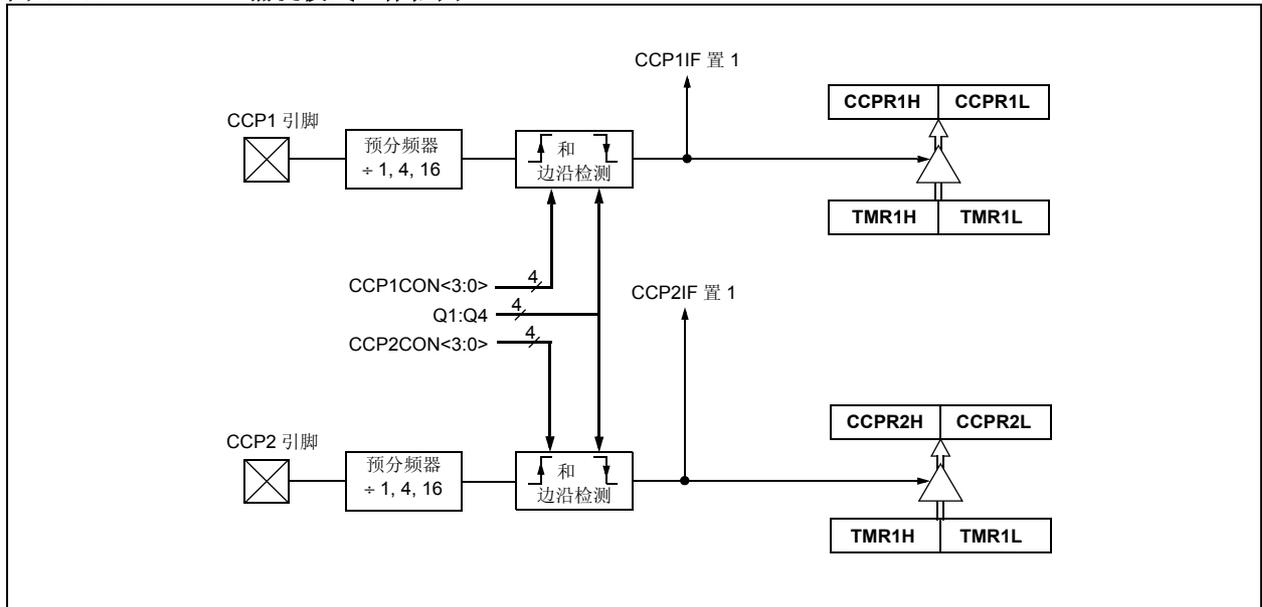
在捕捉模式下有 4 种预分频器设置：它们作为操作模式的一部分由模式选择位 (CCPxM3:CCPxM0) 指定。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

从一个捕捉预分频器切换到另一个捕捉预分频器可能会产生中断。但不会将预分频计数器清零。因此第一个捕捉可能来自一个非零的预分频器。例 13-1 所示为在捕捉预分频器之间进行切换的建议方法。该例也将清零预分频计数器且不会产生“错误”中断。

例 13-1: 在捕捉预分频器之间进行切换 (以 CCP2 为例)

```
CLRF   CCP2CON      ; Turn CCP module off
MOVLW  NEW_CAPT_PS  ; Load WREG with the
                    ; new prescaler mode
                    ; value and CCP ON
MOVWF  CCP2CON      ; Load CCP2CON with
                    ; this value
```

图 13-1: 捕捉模式工作框图



PIC18F45J10 系列

13.3 比较模式

在比较模式下，16位CCPRx寄存器的值将不断与TMR1寄存器对的值相比较。当两者匹配时，CCPx引脚可能会出现以下几种情况：

- 驱动输出高电平
- 驱动输出低电平
- 输出电平翻转（高电平变为低电平或低电平变为高电平）
- 保持不变（即，反映 I/O 锁存器的状态）

引脚的动作取决于模式选择位（CCPxM3:CCPxM0）的值。同时，中断标志位 CCPxIF 置 1。

13.3.1 CCP 引脚配置

用户必须通过将相应的 TRIS 位清零，将 CCPx 引脚配置为输出。

注： 清零 CCP2CON 寄存器会将 RB3 或 RC1 比较输出锁存器（取决于器件配置）强制为默认的低电平。这不是 PORTB 或 PORTC I/O 数据锁存器。

13.3.2 TIMER1 模式选择

如果 CCP 模块工作在比较模式下，Timer1 必须运行在定时器模式或同步计数器模式。在异步计数器模式下，可能无法进行比较操作。

13.3.3 软件中断模式

当选择了产生软件中断模式时（CCPxM3:CCPxM0 = 1010），对应的CCPx引脚不受影响。如果中断被允许并把CCPxIE位置1，只产生一个CCP中断。

13.3.4 特殊事件触发器

两个 CCP 模块均配备了一个特殊事件触发器。这是比较模式产生的内部硬件信号，用于触发其他模块的操作。通过选择比较特殊事件触发模式（CCPxM3:CCPxM0 = 1011），使能特殊事件触发器。

对于任何一个 CCP 模块，无论当前使用哪一个定时器作为模块时基，特殊事件触发器将复位定时器寄存器对。这样就允许 CCPRx 寄存器作为两个定时器的可编程周期寄存器。

CCP2 的特殊事件触发器还能启动 A/D 转换。要实现此功能，A/D 转换器必须已被使能。

图 13-2: 比较模式工作框图

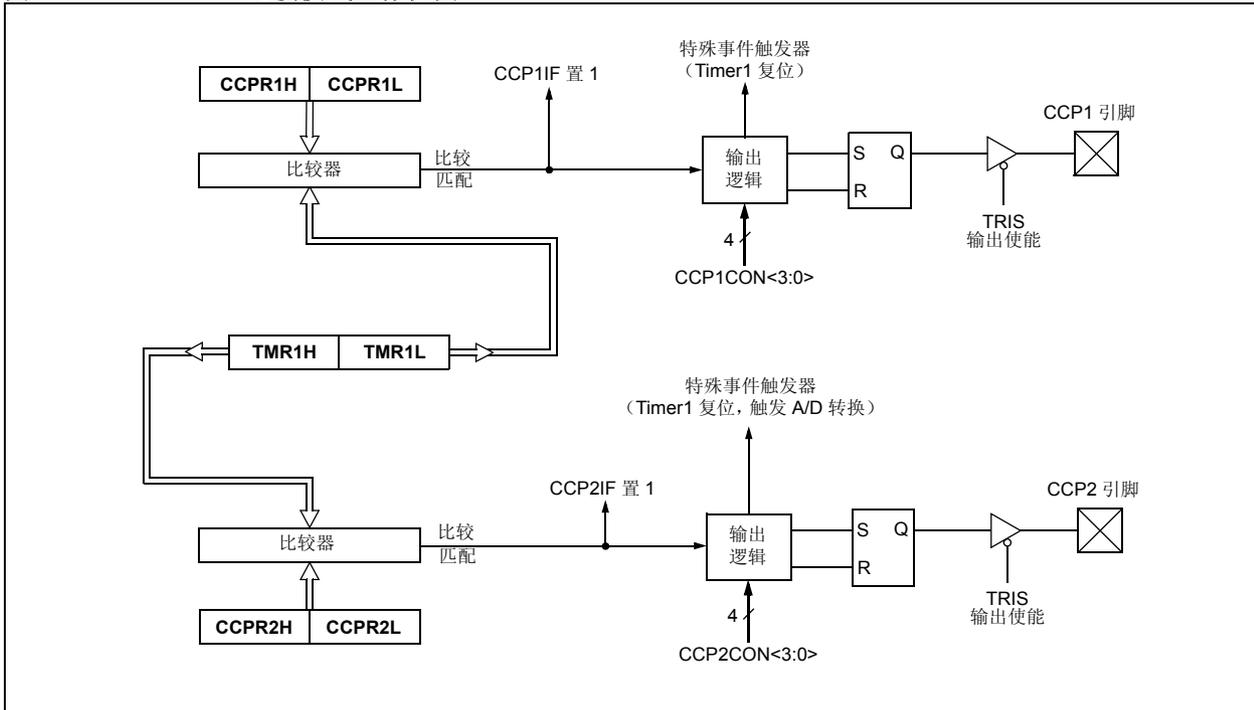


表 13-3: 与捕捉、比较和 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
RCON	IPEN	—	—	RI	TO	PD	POR	BOR	42
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
TRISB	PORTB 数据方向控制寄存器								46
TRISC	PORTC 数据方向控制寄存器								46
TMR1L	Timer1 寄存器的低字节								44
TMR1H	Timer1 寄存器的高字节								44
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \bar{C}	TMR1CS	TMR1ON	44
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								45
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								45
CCP1CON	P1M1 ⁽¹⁾	P1M0 ⁽¹⁾	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	45
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								45
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								45
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	45

图注: — = 未用 (读为 0)。捕捉 / 比较或 Timer1 未使用阴影单元。

注 1: 28 引脚器件不存在此位, 应该被读为 0。

PIC18F45J10 系列

13.4 PWM 模式

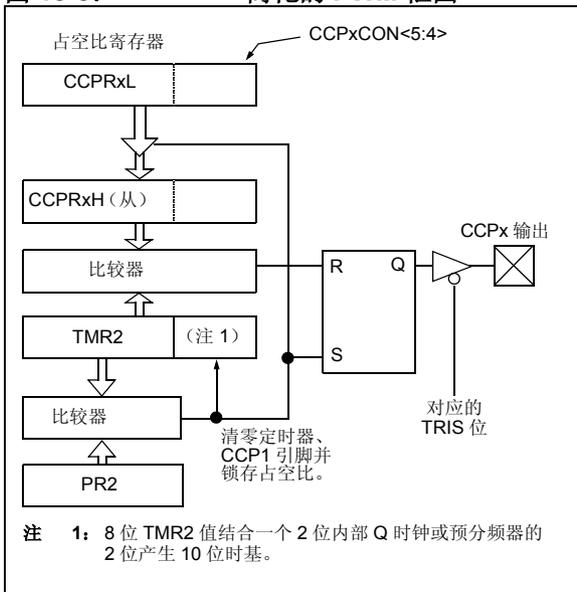
在脉宽调制 (Pulse-Width Modulation, PWM) 模式下, CCPx 引脚可输出分辨率高达 10 位的 PWM 信号。由于 CCP2 引脚与 PORTB 或 PORTC 数据锁存器复用, 必须清零相应的 TRIS 位才能使 CCP2 引脚成为输出引脚。

注: 清零 CCP2CON 寄存器会将 RB3 或 RC1 输出锁存器 (取决于器件配置) 强制为默认的低电平。这不是 PORTB 或 PORTC I/O 数据锁存器。

图 13-3 所示为 PWM 模式下 CCP 模块的简化框图。

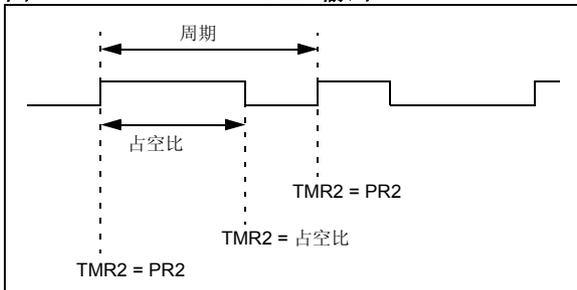
如需了解设置 CCP 模块以进行 PWM 操作的详细步骤, 请参见第 13.4.4 节“设置 PWM 操作”。

图 13-3: 简化的 PWM 框图



一个 PWM 输出 (图 13-4) 包含一个时基 (周期) 和一段输出高电平的时间 (占空比)。PWM 的频率是周期的倒数 (1/周期)。

图 13-4: PWM 输出



13.4.1 PWM 周期

PWM 周期是通过写 PR2 寄存器来指定的。可以使用以下公式计算 PWM 周期:

公式 13-1:

$$\text{PWM 周期} = [(\text{PR2}) + 1] \cdot 4 \cdot \text{Tosc} \cdot (\text{TMR2 预分频值})$$

PWM 频率定义为 $1 / [\text{PWM 周期}]$ 。

当 TMR2 等于 PR2 时, 在下一个递增计数周期中会发生以下 3 个事件:

- TMR2 被清零
- CCPx 引脚被置 1 (例外情况: 如果 PWM 占空比 = 0%, CCPx 引脚将不被置 1)
- PWM 占空比从 CCPRxL 被锁存到 CCPRxH

注: 不使用 Timer2 后分频器 (参见第 12.0 节“Timer2 模块”) 来确定 PWM 频率。使用后分频器时, 其伺服更新速率的频率可与 PWM 输出频率不同。

13.4.2 PWM 占空比

PWM 占空比可通过写入 CCPRxL 寄存器和 CCPxCON<5:4> 位来指定。其最高分辨率可达 10 位。CCPRxL 保存占空比高 8 位, CCPxCON<5:4> 保存低 2 位。CCPRxL:CCPxCON<5:4> 代表这个 10 位值。计算占空比的公式如下:

公式 13-2:

$$\text{PWM 占空比} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot \text{Tosc} \cdot (\text{TMR2 预分频值})$$

可以在任何时候写入 CCPRxL 和 CCPxCON<5:4>, 但直到 PR2 和 TMR2 中的值匹配 (即, 当周期结束时) 时, 占空比的值才被锁存到 CCPRxH 中。在 PWM 模式下, CCPRxH 是只读寄存器。

CCPRxH 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲结构极其重要，可以避免在 PWM 切换占空比的时候产生毛刺。

当 CCPRxH 和 2 位锁存器的值与 TMR2 加上内部 2 位 Q 时钟或 TMR2 预分频器的 2 位的值匹配时，CCPx 引脚被清零。

对于给定的 PWM 频率，其最大分辨率（位）可以由以下公式计算：

公式 13-3:

$$\text{PWM 分辨率 (最高)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ 位}$$

注：如果 PWM 的占空比值大于 PWM 周期，CCP2 引脚将不会被清零。

表 13-4: 40 MHz 下的 PWM 频率和分辨率示例

PWM 频率	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
定时器预分频值（1、4 或 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最大分辨率（位）	10	10	10	8	7	6.58

13.4.3 PWM 自动关闭（仅 CCP1）

28 引脚器件的 CCP1 也可以使用增强型 CCP 模块的 PWM 自动关闭功能。在第 14.4.7 节“增强型 PWM 的自动关闭功能”中对此功能的操作进行了详细的讨论。

CCP2 不可以使用自动关闭功能。

13.4.4 设置 PWM 操作

在为 PWM 操作配置 CCP 模块时应该遵循以下步骤：

1. 通过写 PR2 寄存器来设置 PWM 周期。
2. 通过写 CCPRxL 寄存器和 CCPxCON<5:4> 位来设置 PWM 占空比。
3. 通过清零相应的 TRIS 位将 CCPx 引脚配置为输出。
4. 通过写 T2CON 来设置 TMR2 预分频值并使能 Timer2。
5. 配置 CCPx 模块以进行 PWM 操作。

PIC18F45J10 系列

表 13-5: 与 PWM 和 TIMER2 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
RCON	IPEN	—	—	RI	TO	PD	POR	BOR	42
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
TRISB	PORTB 数据方向控制寄存器								46
TRISC	PORTC 数据方向控制寄存器								46
TMR2	Timer2 寄存器								44
PR2	Timer2 周期寄存器								44
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	44
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								45
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								45
CCP1CON	P1M1 ⁽¹⁾	P1M0 ⁽¹⁾	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	45
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								45
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								45
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	45
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 ⁽¹⁾	PSSBD0 ⁽¹⁾	45
ECCP1DEL	PRSEN	PDC6 ⁽¹⁾	PDC5 ⁽¹⁾	PDC4 ⁽¹⁾	PDC3 ⁽¹⁾	PDC2 ⁽¹⁾	PDC1 ⁽¹⁾	PDC0 ⁽¹⁾	45

图注: — = 未用, 读为 0。PWM 或 Timer2 不使用阴影单元。

注 1: 28 引脚器件不存在这些位, 应该被读为 0。

14.0 增强型捕捉 / 比较 / PWM (ECCP) 模块

注： 只有 40/44 引脚器件中才有 ECCP 模块。

在 PIC18F44J10/45J10 器件中，ECCP1 为带有增强的 PWM 功能的标准 CCP 模块。这些增强的功能包括提供 2 路或 4 路输出通道、用户可选的极性、死区控制以及

自动关闭和重新启动。第 14.4 节“增强型 PWM 模式”中有对增强功能的详细讨论。ECCP 模块的捕捉、比较和单输出 PWM 功能与标准 CCP 模块中描述的相同。

增强型 CCP 模块的控制寄存器如寄存器 14-1 所示。它与 PIC18F24J10/25J10 器件中的 CCP1CON 寄存器的不同之处在于其高 2 位用来控制 PWM 功能。

寄存器 14-1: CCP1CON 寄存器 (ECCP1 模块, 40/44 引脚器件)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7				bit 0			

- bit 7-6 P1M1:P1M0:** 增强型 PWM 输出配置位
 如果 CCP1M3:CCP1M2 = 00、01 或 10:
 xx = P1A 被分配为捕捉 / 比较输入 / 输出; P1B、P1C 和 P1D 被分配为端口引脚
 如果 CCP1M3:CCP1M2 = 11:
 00 = 单输出: P1A 调制输出; P1B、P1C 和 P1D 被分配为端口引脚
 01 = 全桥正向输出: P1D 调制输出; P1A 有效; P1B 和 P1C 无效
 10 = 半桥输出: P1A 和 P1B 为带死区控制的调制输出; P1C 和 P1D 被分配为端口引脚
 11 = 全桥反向输出: P1B 调制输出; P1C 有效; P1A 和 P1D 无效
- bit 5-4 DC1B1:DC1B0:** PWM 占空比 bit 1 和 bit 0
捕捉模式:
 未用。
比较模式:
 未用。
PWM 模式:
 这两位是 10 位 PWM 占空比的低 2 位。占空比的高 8 位在 CCPR1L 中。
- bit 3-0 CCP1M3:CCP1M0:** 增强型 CCP 模式选择位
 0000 = 捕捉 / 比较 / PWM 关闭 (复位 ECCP 模块)
 0001 = 保留
 0010 = 比较模式, 匹配时输出电平翻转
 0011 = 捕捉模式
 0100 = 捕捉模式, 在每个下降沿发生捕捉
 0101 = 捕捉模式, 在每个上升沿发生捕捉
 0110 = 捕捉模式, 每 4 个上升沿发生捕捉
 0111 = 捕捉模式, 每 16 个上升沿发生捕捉
 1000 = 比较模式, CCP1 引脚初始化为低电平, 比较匹配时输出高电平 (置位 CCP1IF)
 1001 = 比较模式, CCP1 引脚初始化为高电平, 比较匹配时输出低电平 (置位 CCP1IF)
 1010 = 比较模式, 仅产生软件中断, CCP1 引脚恢复到 I/O 状态
 1011 = 比较模式, 触发特殊事件 (ECCP 复位 TMR1, 将 CCP1IF 位置 1)
 1100 = PWM 模式; P1A 和 P1C 为高电平有效; P1B 和 P1D 也为高电平有效
 1101 = PWM 模式; P1A 和 P1C 为高电平有效; P1B 和 P1D 为低电平有效
 1110 = PWM 模式; P1A 和 P1C 为低电平有效; P1B 和 P1D 为高电平有效
 1111 = PWM 模式; P1A 和 P1C 为低电平有效; P1B 和 P1D 也为低电平有效

图注:

R = 可读位 W = 可写位 U = 未用位, 读为 0
 -n = 上电复位时的值 1 = 置 1 0 = 清零 x = 未知

PIC18F45J10 系列

除了通过 CCP1CON 寄存器和 ECCP1AS 寄存器可获得扩展模式以外，ECCP 模块还有一个与增强型 PWM 操作和自动关闭功能相关的寄存器。该寄存器是：

- ECCP1DEL（死区延时寄存器）

14.1 ECCP 输出和配置

每个增强型 CCP 模块至多有 4 路 PWM 输出，这取决于所选定的操作模式。这些输出，P1A 至 P1D，与 PORTC 和 PORTD 上的 I/O 引脚复用。输出是否有效取决于所选定的 ECCP 操作模式。表 14-1 总结了引脚的分配情况。

要将 I/O 引脚配置为 PWM 输出，必须通过设置 P1M1:P1M0 以及 CCP1M3:CCP1M0 位来选择适当的 PWM 模式。还必须将对应端口引脚的相应 TRISC 和 TRISD 方向位设置为输出。

14.1.1 ECCP 模块和定时器资源

与标准 CCP 模块相同，ECCP 模块可以使用 Timer1 或 Timer2，具体选用哪个定时器取决于所选定的模式。Timer1 在捕捉或比较模式下用于该模块，而 Timer2 则在 PWM 模式下用于该模块。标准 CCP 模块与增强型 CCP 模块的互连方式与标准 CCP 模块的相同。第 13.1.1 节“CCP 模块和定时器资源”中提供了更多关于定时器资源的详细信息。

14.2 捕捉和比较模式

除了要在下面讨论的特殊事件触发器的操作，ECCP 模块的捕捉和比较模式操作与 CCP2 相同。第 13.2 节“捕捉模式”和第 13.3 节“比较模式”将详细讨论这方面的内容。在 28 引脚器件和 40/44 引脚器件之间进行移植时，不需要作任何更改。

14.2.1 特殊事件触发器

ECCP1 的特殊事件触发器输出将复位 TMR1 寄存器对。这将使 CCPR1 寄存器被用作 Timer1 的 16 位可编程周期寄存器。

14.3 标准 PWM 模式

当被配置为单输出模式时，ECCP 模块的工作方式与第 13.4 节“PWM 模式”中描述的 PWM 模式下的标准 CCP 模块相同。这种模式有时也被称为“兼容的 CCP”模式，如表 14-1 所示。

注： 在设置单输出 PWM 操作时，用户可以随意使用第 13.4.4 节“设置 PWM 操作”或第 14.4.9 节“设置 PWM 操作”中说明的步骤。后者更加常用但是只适用于单输出或多输出 PWM。

表 14-1: 各种 ECCP1 模式的引脚分配

ECCP 模式	CCP1CON 配置	RC2	RD5	RD6	RD7
所有 40/44 引脚器件：					
兼容的 CCP	00xx 11xx	CCP1	RD5/PSP5	RD6/PSP6	RD7/PSP7
双输出 PWM	10xx 11xx	P1A	P1B	RD6/PSP6	RD7/PSP7
四输出 PWM	x1xx 11xx	P1A	P1B	P1C	P1D

图注： x = 任意值。阴影单元表示在给定模式下 ECCP1 不使用的引脚。

14.4 增强型 PWM 模式

增强型 PWM 模式提供了更多的 PWM 输出选项以适应范围更广的控制应用。该模块是标准 CCP 模块的向下兼容版本，提供至多 4 路输出，即 P1A 到 P1D。用户还能够选择信号的极性（高电平有效或低电平有效）。通过分别设置 CCP1CON 寄存器的 P1M1:P1M0 和 CCP1M3:CCP1M0 位可配置模块的输出模式和极性。

图 14-1 所示为 PWM 操作的简化框图。所有的控制寄存器都是双重缓冲的，并且在一个新的 PWM 周期的开始时刻（Timer2 复位时的周期边界）被装载以防止在输出上出现毛刺。但 PWM 延迟寄存器 ECCP1DEL 例外，该寄存器在占空比边界或周期边界（选择两者中首先出现的那个）被装载。由于缓冲，模块将不会立即启动，而要等到分配的定时器复位。这意味着增强型 PWM 波形并不完全与标准的 PWM 波形吻合，而是偏移一个完整的指令周期（4 T_{osc}）。

如前所述，用户必须为输出手动配置相应的 TRIS 位。

14.4.1 PWM 周期

PWM 周期可以通过写 PR2 寄存器来指定。可使用以下公式来计算 PWM 周期。

公式 14-1:

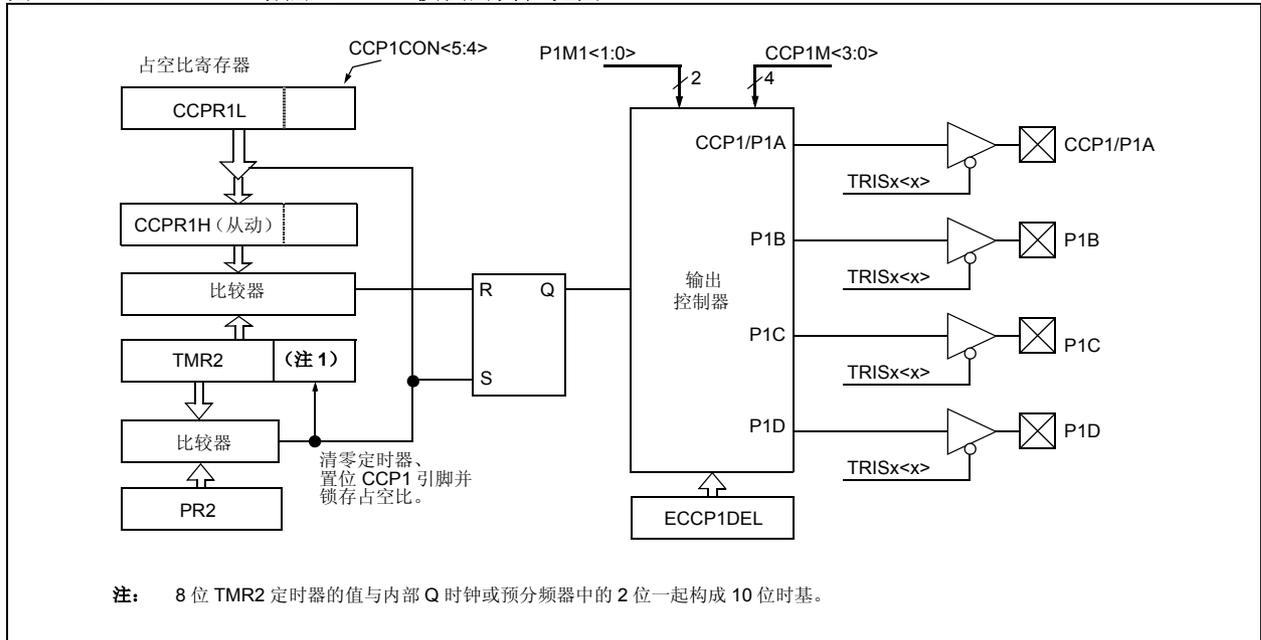
$$\text{PWM 周期} = \frac{[(PR2) + 1] \cdot 4 \cdot T_{osc}}{(\text{TMR2 预分频值})}$$

PWM 频率定义为 1 / [PWM 周期]。当 TMR2 等于 PR2 时，在下一个递增计数周期中会发生以下三个事件：

- TMR2 被清零
- CCP1 引脚被置 1（如果 PWM 占空比 = 0%，CCP1 引脚将不会被置 1）
- PWM 占空比从 CCPR1L 复制到 CCPR1H

注： Timer2 后分频器不被用来确定 PWM 频率（见第 12.0 节“Timer2 模块”）。使用后分频器可用不同于 PWM 输出频率的频率进行相应的数据更新。

图 14-1: 增强型 PWM 模块的简化框图



PIC18F45J10 系列

14.4.2 PWM 占空比

通过写 CCPR1L 寄存器和 CCP1CON<5:4> 位来指定 PWM 占空比，其最高分辨率可达 10 位。CCPR1L 寄存器包含占空比的高 8 位，而 CCP1CON<5:4> 包含低 2 位。由 CCPR1L:CCP1CON<5:4> 表示完整的 10 位值。计算占空比的公式如下：

公式 14-2:

$$\text{PWM 占空比} = \frac{(\text{CCPR1L:CCP1CON<5:4>}) \cdot \text{Tosc}}{\text{TMR2 预分频值}}$$

可以在任何时候写入 CCPR1L 和 CCP1CON<5:4>，但是在 PR2 和 TMR2 发生匹配（即周期结束）前占空比值不会被锁存到 CCPR1H 中。在 PWM 模式中，CCPR1H 是只读寄存器。

CCPR1H 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲结构非常重要，可以避免在 PWM 工作过程中产生毛刺。当 CCPR1H 和 2 位锁存值与 TMR2（连有内部 2 位 Q 时钟或 TMR2 预分频器值中的 2 位）匹配时，CCP1 引脚被清零。对于给定的 PWM 频率，其最高 PWM 分辨率（位）由以下公式计算：

公式 14-3:

$$\text{PWM 分辨率 (最高)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ 位}$$

注： 如果 PWM 占空比的值大于 PWM 周期，CCP1 引脚将不会被清零。

14.4.3 PWM 输出配置

CCP1CON 寄存器中的 P1M1:P1M0 位可以实现以下 4 种配置：

- 单输出
- 半桥输出
- 全桥输出，正向模式
- 全桥输出，反向模式

单输出模式是在第 14.4 节“增强型 PWM 模式”中讨论的标准 PWM 模式。在接下来的各节中将详细介绍半桥和全桥输出模式。

图 14-2 中汇总了各种配置下输出的关系。

表 14-2: 40 MHz 时的 PWM 频率和分辨率示例

PWM 频率	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
定时器预分频值（1、4 或 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最高分辨率（位）	10	10	10	8	7	6.58

图 14-2: PWM 输出关系 (高电平有效状态)

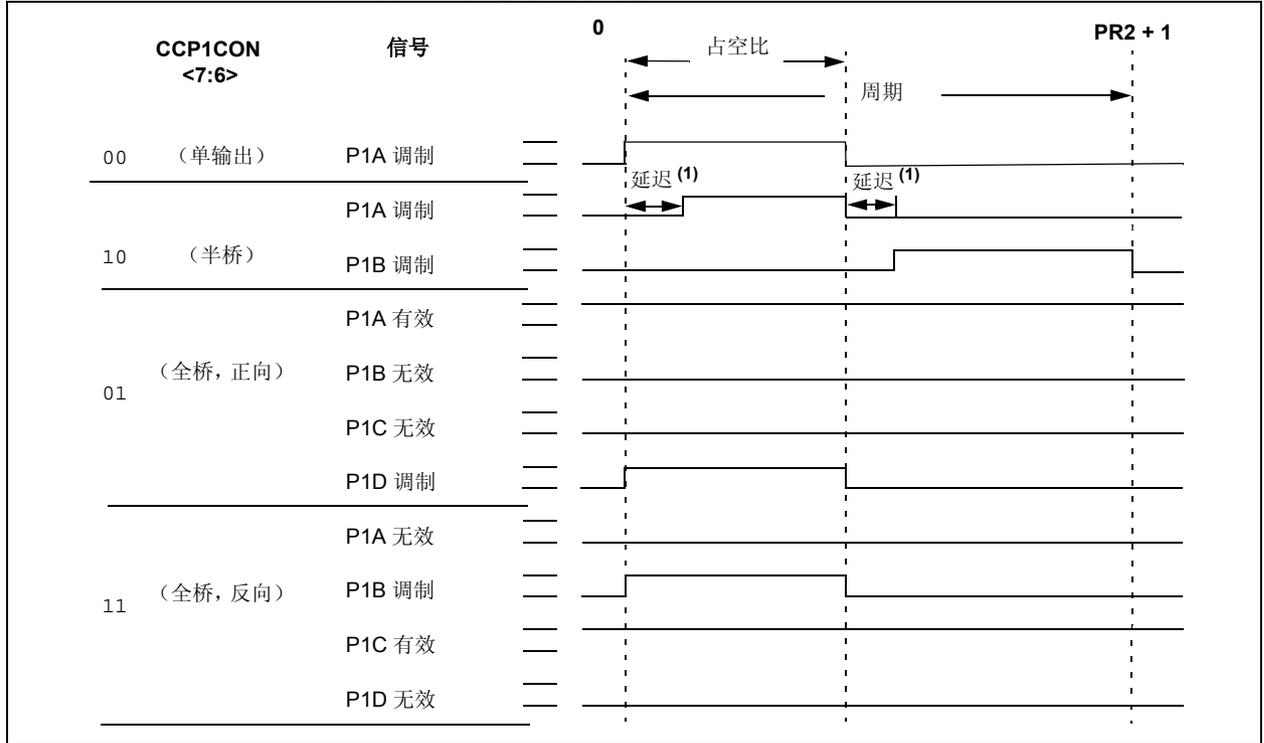
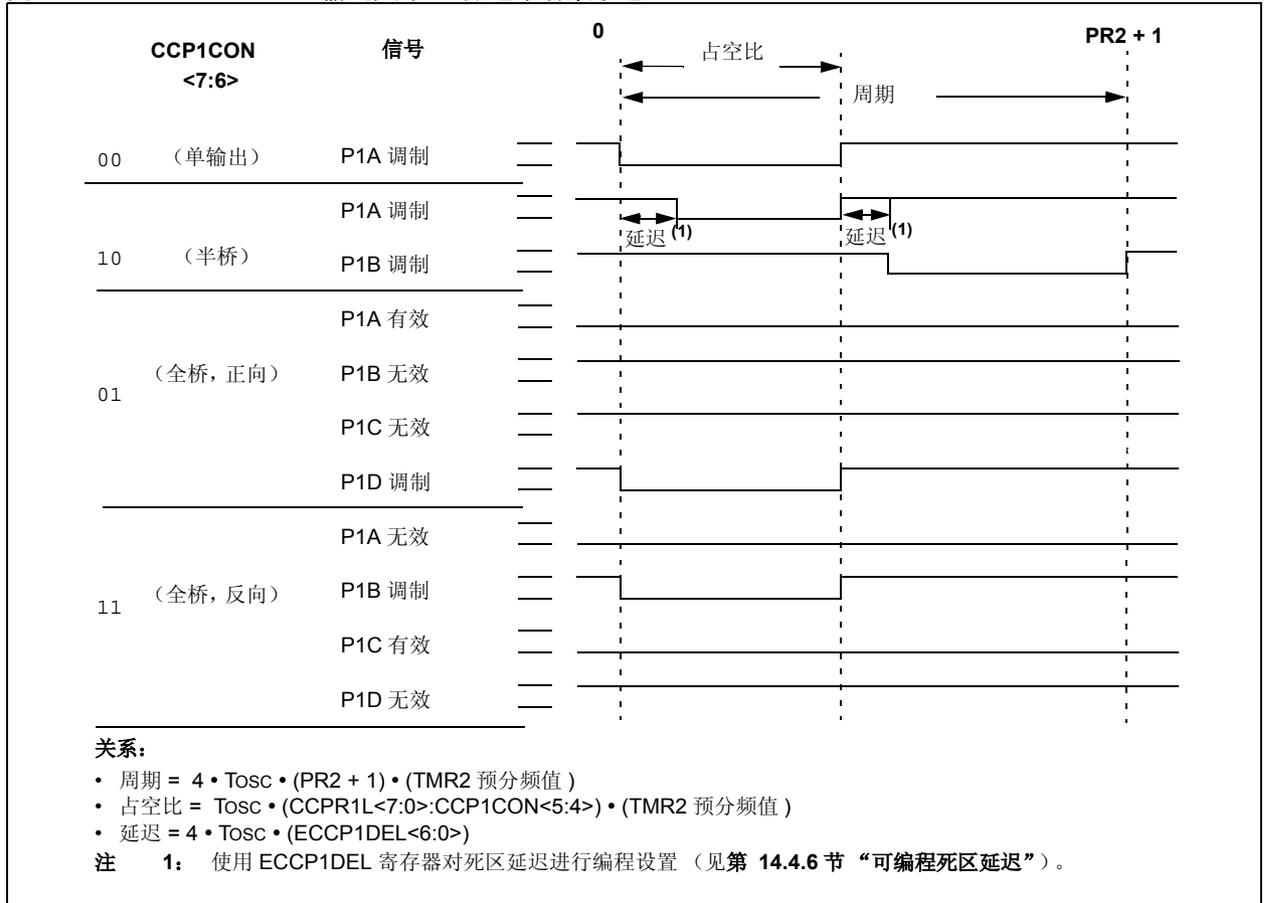


图 14-3: PWM 输出关系 (低电平有效状态)



PIC18F45J10 系列

14.4.4 半桥模式

在半桥输出模式下，两个引脚用作输出端来驱动推挽式负载。PWM 输出信号在 P1A 引脚上输出，而互补的 PWM 输出信号在 P1B 引脚上输出（图 14-4）。如图 14-5 所示，此模式可用于半桥应用，或那些使用 2 个 PWM 信号来调制 4 个电源开关的全桥应用。

在半桥输出模式下，可编程的死区延迟用来防止在半桥电源器件中产生直通电流。PDC6:PDC0 位的值用来设置输出被驱动为有效之前的指令周期数。如果该值大于占空比，在整个周期内对应的输出将保持无效。有关死区延迟操作的更多详细信息，请参见第 14.4.6 节“可编程死区延迟”。

由于 P1A 和 P1B 输出与 PORTC<2> 和 PORTD<5> 数据锁存器复用，必须清零 TRISC<2> 和 TRISD<5> 位将 P1A 和 P1B 配置为输出。

图 14-4: 半桥 PWM 输出

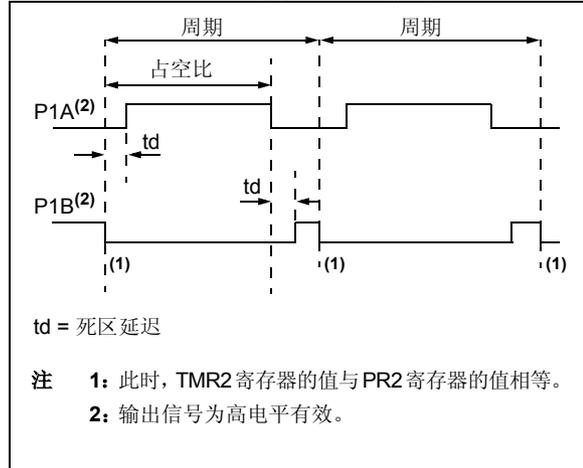
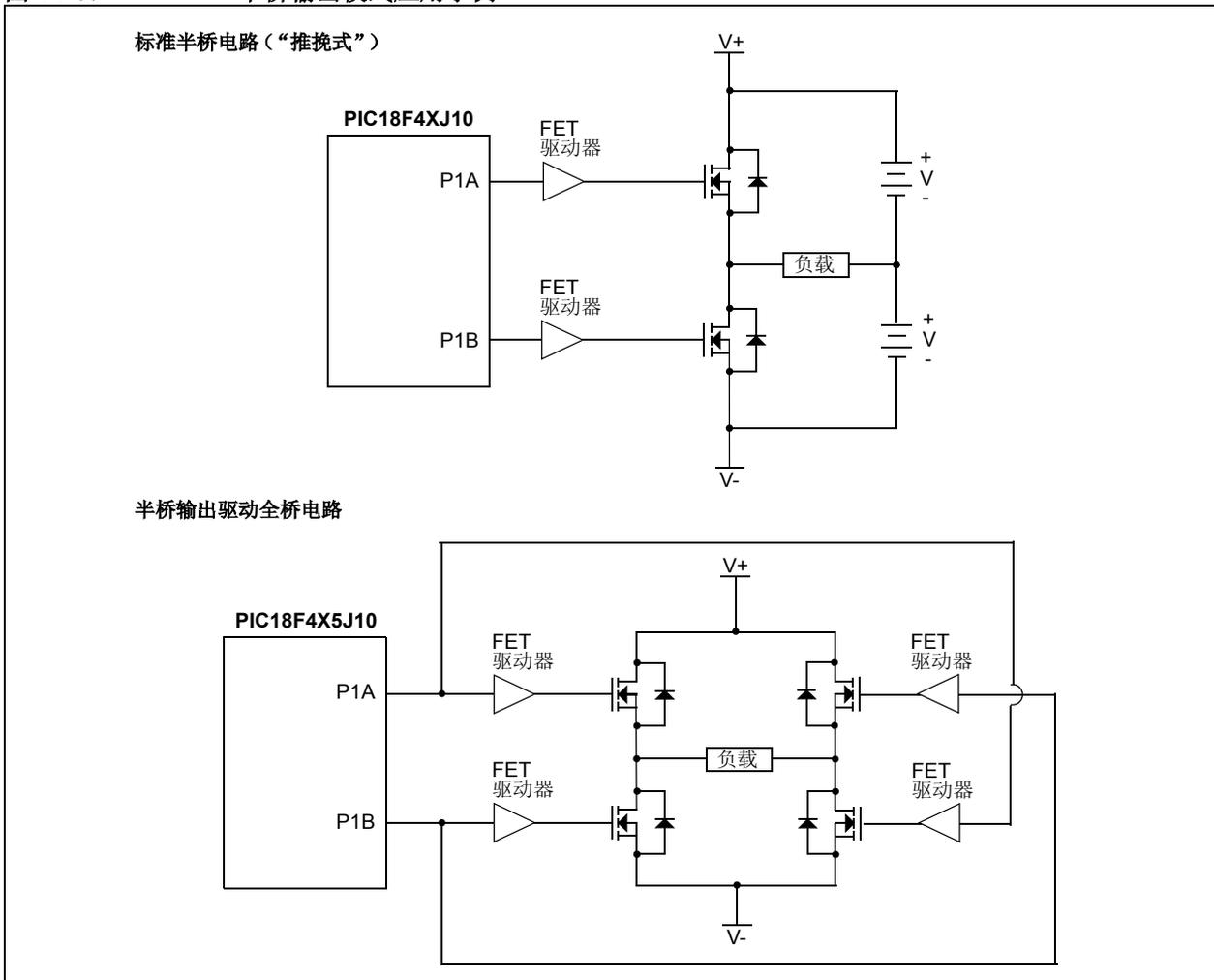


图 14-5: 半桥输出模式应用示例

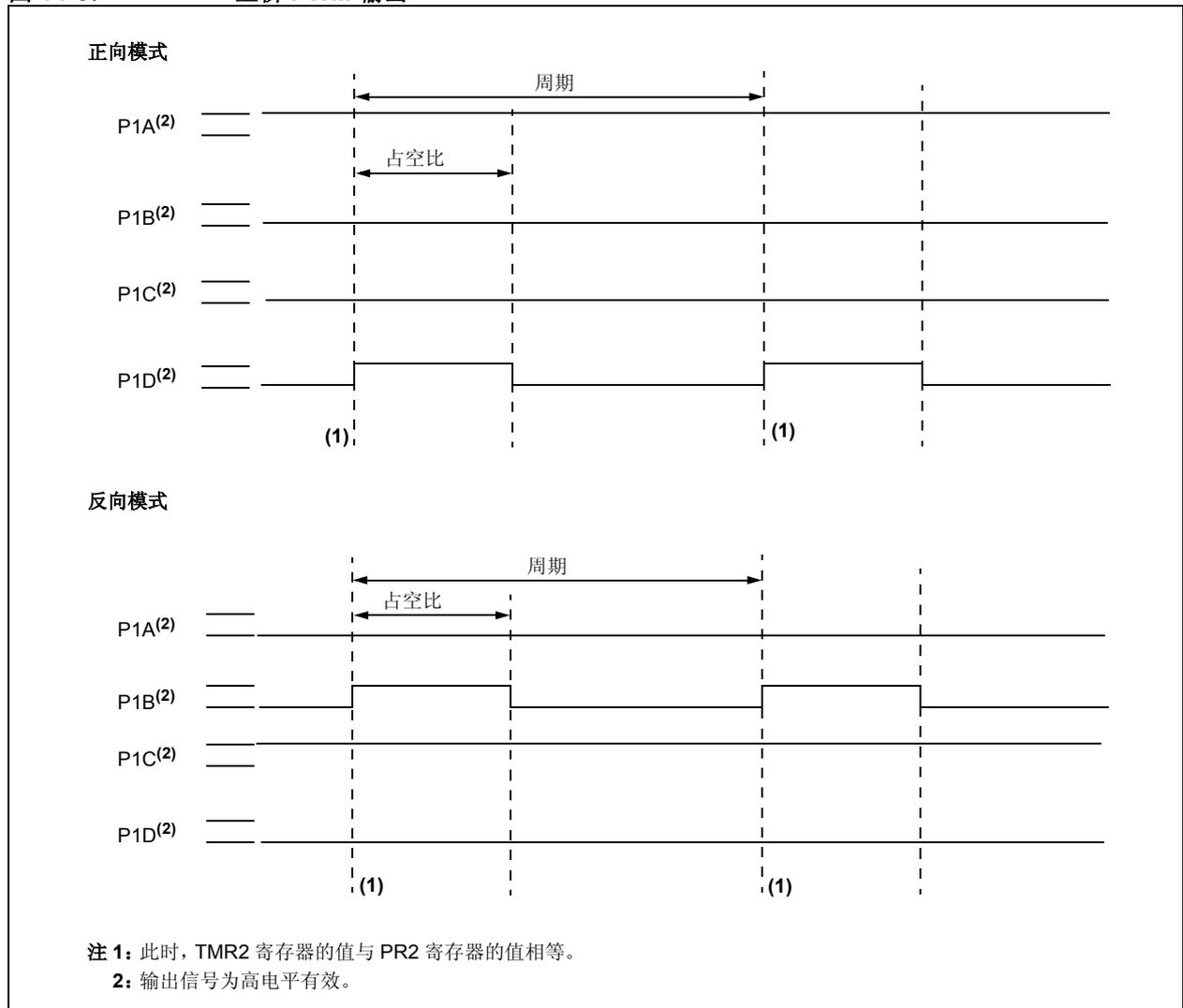


14.4.5 全桥模式

在全桥输出模式下，4 个引脚被用作输出；但是，在任一时刻只能有 2 个输出同时有效。在正向模式下，引脚 P1A 一直有效而引脚 P1D 为调制输出。在反向模式下，引脚 P1C 一直有效而引脚 P1B 为调制输出。这些在图 14-6 中进行了说明。

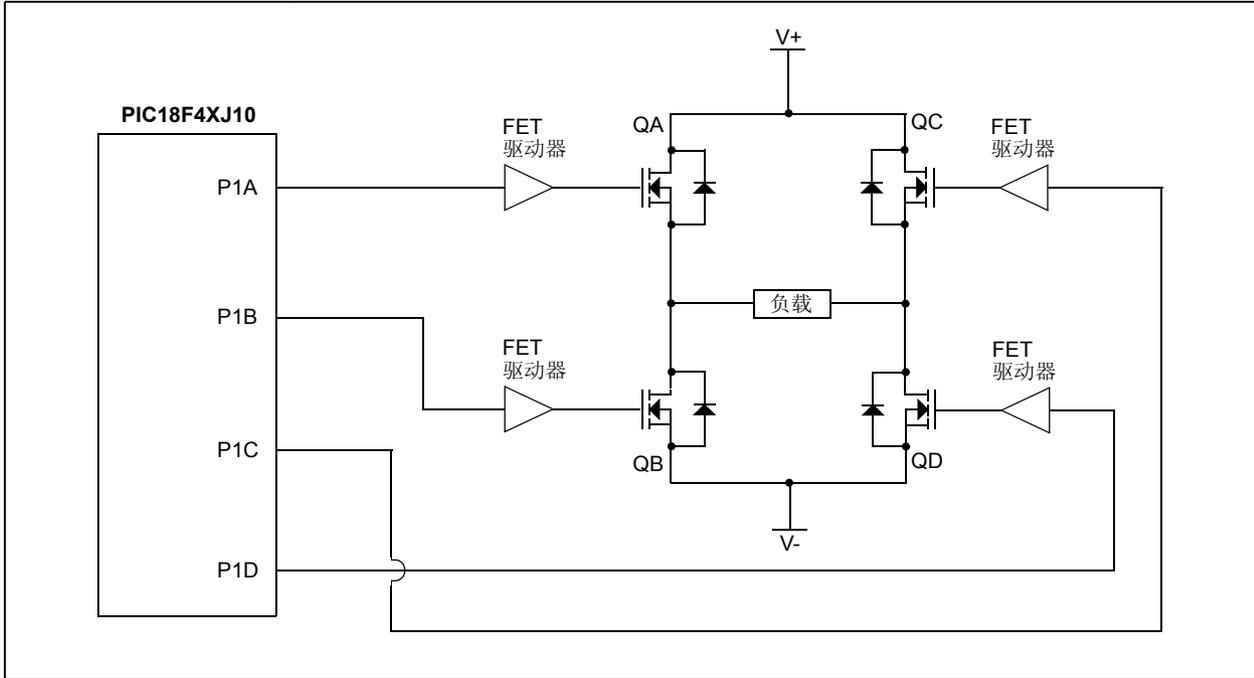
P1A、P1B、P1C 和 P1D 输出与 PORTC<2> 和 PORTD<7:5> 数据锁存器复用。必须清零 TRISC<2> 和 TRISD<7:5> 位使 P1A、P1B、P1C 和 P1D 作为输出引脚。

图 14-6: 全桥 PWM 输出



PIC18F45J10 系列

图 14-7: 全桥应用示例



14.4.5.1 全桥模式下的方向更改

在全桥输出模式下，CCP1CON 寄存器中的 P1M1 位使用户能控制负载中电流为正向还是反向。当应用固件更改此方向控制位时，模块将在下一个 PWM 周期采用新的方向。

就在当前 PWM 周期结束之前，调制输出（P1B 和 P1D）被置于无效状态，而非调制输出（P1A 和 P1C）被切换到以相反的方向驱动负载。这发生在下一个 PWM 周期开始前的一段时间间隔内（ $4 T_{osc} \cdot (\text{Timer2 预分频值})$ ）。Timer2 预分频器的分频值将是 1、4 或 16，这取决于 T2CKPS1:T2CKPS0 位（T2CON<1:0>）的值。从切换非调制输出到下一个周期开始之间的这段时间内，调制输出（P1B 和 P1D）保持无效状态。此关系如图 14-8 所示。

注意在全桥输出模式下，ECCP1 模块不提供任何死区延迟。通常，由于在任何时间只调制一个输出，所以不需要死区延迟。然而，有一种情形可能需要死区延迟，即，当以下两个条件都为真时的情况：

1. 当输出信号的占空比接近或等于 100% 时，PWM 输出的方向发生改变。
2. 电源开关（包括电源器件和驱动电路）的关断时间大于导通时间。

图 14-9 所示为占空比接近 100% 时，PWM 方向从正向更改为反向的示例。在时间 t_1 时，输出 P1A 和 P1D 变为无效，而输出 P1C 变为有效。在此示例中，由于电源器件的关断时间比导通时间长，直通电流可能在时间段“t”内流过电源器件 QC 和 QD（见图 14-7）。如果 PWM 方向从反向更改为正向，电源器件 QA 和 QB 将出现相同的现象。

如果需要在高占空比情况下更改 PWM 方向，必须满足以下要求之一：

1. 在更改方向前的那个 PWM 周期减小 PWM 的占空比。
2. 使用可使开关元件的关断速度比导通速度更快的开关驱动器。

可能还存在防止直通电流的其他方法。

图 14-8: PWM 方向更改

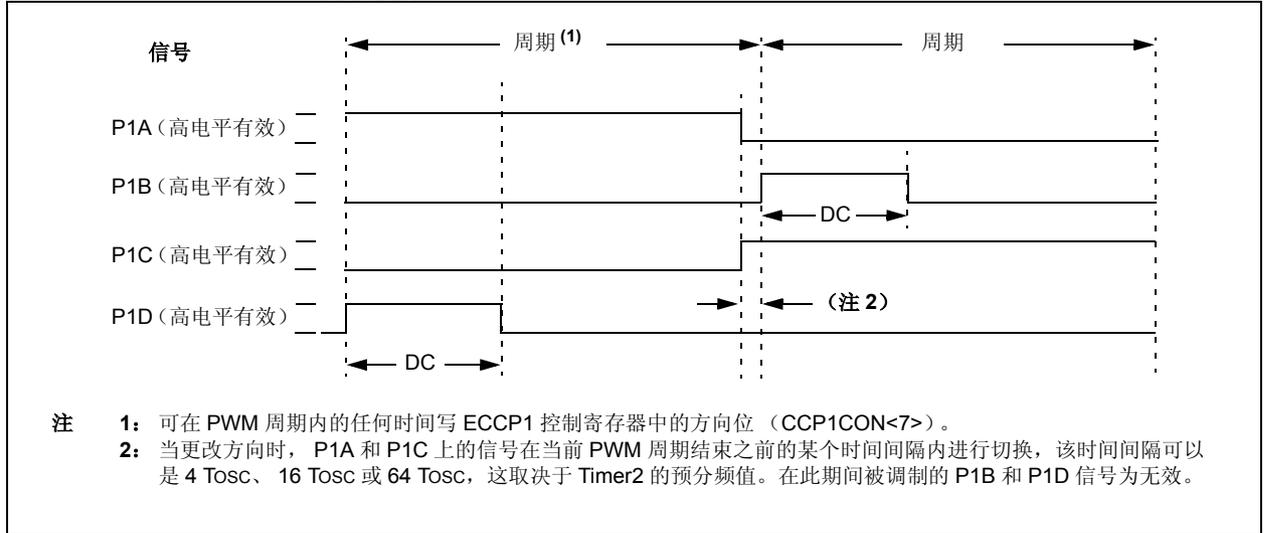
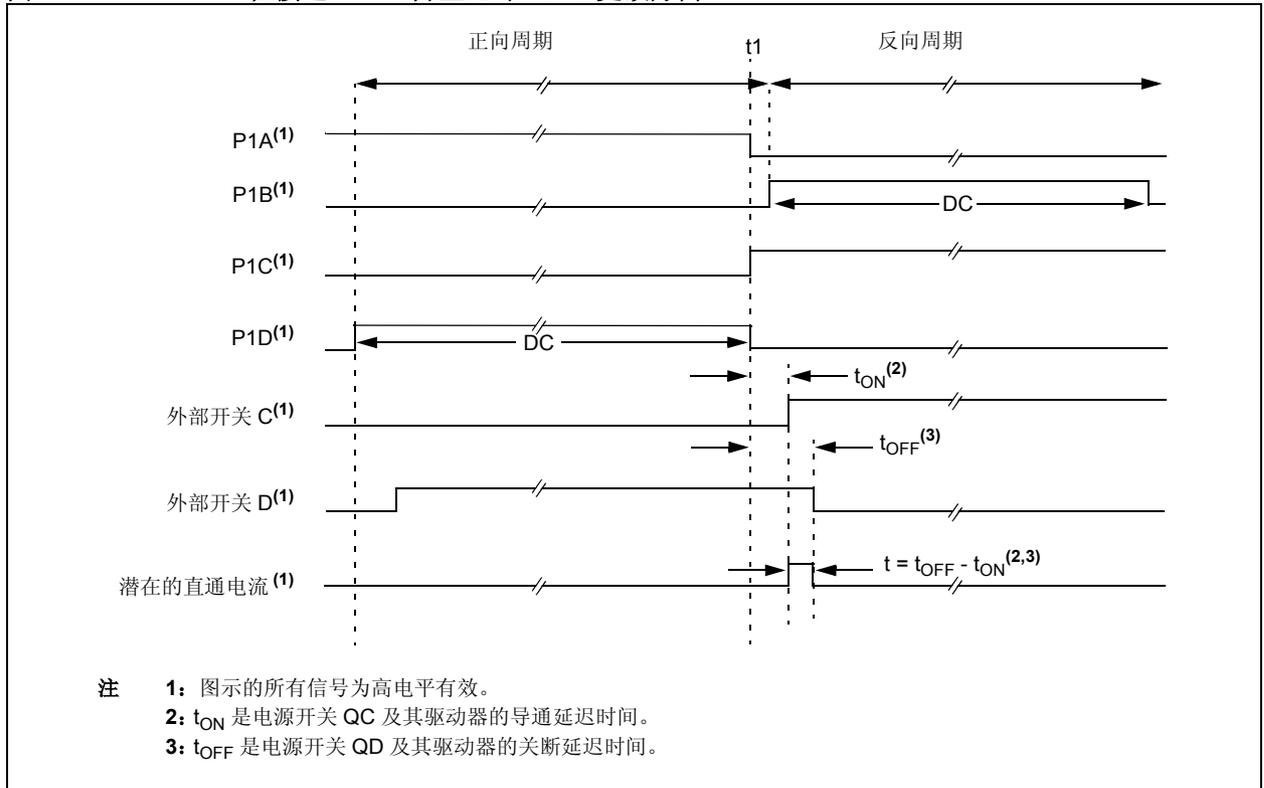


图 14-9: 在接近 100% 占空比时 PWM 更改方向



PIC18F45J10 系列

14.4.6 可编程死区延迟

注： 在带有标准 CCP 模块的 28 引脚器件中没有实现可编程死区延迟功能。

在半桥应用中，模块一直以 PWM 频率的调制信号驱动电源开关，关闭电源开关通常比打开它需要更多的时间。如果上方的电源开关和下方的电源开关同时切换（一个打开，另一个关闭），两个开关可能会在一段很短的时间内都处于打开状态，直到一个开关完全关闭为止。在这很短的间隔内，很大的电流（*直通电流*）可能流过两个电源开关，从而导致半桥供电电路短路。为了避免在切换期间流过这种潜在的破坏性直通电流，一般延迟打开其中的一个电源开关直到另一个开关完全关闭。

在半桥输出模式下，可数字编程的死区延迟可用来避免直通电流破坏电源开关。该延迟在信号从非有效状态到有效状态转换时发生。参见图 14-4。ECCP1DEL 寄存器（寄存器 14-2）的 PDC6:PDC0 位以单片机指令周期为单位设置延迟时间（ T_{CY} 或 $4 T_{OSC}$ ）。由于标准 CCP 模块不支持半桥工作方式，因此在 28 引脚器件中不使用这些位。

14.4.7 增强型 PWM 的自动关闭功能

当 ECCP1 被编程为任何一种增强型 PWM 模式时，可以将有效输出引脚配置为自动关闭。当发生关闭事件时，自动关闭功能会立即将增强型 PWM 输出引脚置于定义的关闭状态。

关闭事件可以由两个比较器模块中的任意一个、故障输入引脚（FLT0）上的低电平或以上 3 者的任意组合触发。比较器可以用来监视电压输入，该电压与桥式电路中监视到的电流成正比。如果电压超过门限值，比较器将切换状态并触发关闭。此外，FLT0 引脚上的数字低电平信号也能触发关闭。通过不选择任何自动关闭源，可以禁止自动关闭功能。通过使用 ECCPAS2:ECCPAS0 位（ECCP1AS 寄存器的位 <6:4>）选择将使用的自动关闭源。

当关闭事件发生时，输出引脚被异步地置于它们的关闭状态，关闭状态由 PSSAC1:PSSAC0 和 PSSBD1:PSSBD0 位（ECCPAS3:ECCPAS0）指定。每个引脚对（P1A/P1C 和 P1B/P1D）可被设置为驱动高电平、驱动低电平或三态（不驱动）。还需将 ECCPASE 位（ECCP1AS<7>）置 1 以便将增强型 PWM 输出保持在关闭状态。

当关闭事件发生时，ECCPASE 被硬件置 1。如果不能自动重启，当关闭源被清除后，ECCPASE 位将被固件清零。如果使能自动重启，当自动关闭源清除后，ECCPASE 位将被自动清零。

如果当 PWM 周期开始时 ECCPASE 位置 1，PWM 输出将在这整个 PWM 周期内保持关闭状态。当 ECCPASE 位清零时，PWM 输出将在下一个 PWM 周期的开始时恢复正常。

注： 当关闭条件有效时，禁止写 ECCPASE 位。

寄存器 14-2: ECCP1DEL: PWM 死区延迟寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRSEN	PDC6 ⁽¹⁾	PDC5 ⁽¹⁾	PDC4 ⁽¹⁾	PDC3 ⁽¹⁾	PDC2 ⁽¹⁾	PDC1 ⁽¹⁾	PDC0 ⁽¹⁾
bit 7							bit 0

bit 7 **PRSEN:** PWM 重启使能位
 1 = 自动关闭时，一旦关闭事件被清除，ECCPASE 位立即自动清零；PWM 自动重启
 0 = 自动关闭时，必须用软件清零 ECCPASE 位以重启 PWM

bit 6-0 **PDC6:PDC0:** PWM 延迟计数位⁽¹⁾
 延迟时间为 PWM 信号转换到有效的预计时间和实际时间之差，其单位为 $F_{OSC}/4$ ($4 \cdot T_{OSC}$) 周期。

注 1: 在 28 引脚器件中，这些位是保留的，它们始终保持清零。

图注:

R = 可读位 W = 可写位 U = 未用位，读为 0
 -n = 上电复位时的值 1 = 置 1 0 = 清零 x = 未知

寄存器 14-3: ECCP1AS: 增强型捕捉 / 比较 / PWM 自动关闭控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 ⁽¹⁾	PSSBD0 ⁽¹⁾
bit 7						bit 0	

bit 7 **ECCPASE:** ECCP 自动关闭事件状态位
 1 = 已发生了关闭事件; ECCP 输出处于关闭状态
 0 = ECCP 输出正常运行

bit 6-4 **ECCPAS2:ECCPAS0:** ECCP 自动关闭源选择位
 111 = FLT0、比较器 1 或比较器 2
 110 = FLT0 或比较器 2
 101 = FLT0 或比较器 1
 100 = FLT0
 011 = 比较器 1 或 2
 010 = 比较器 2 输出
 001 = 比较器 1 输出
 000 = 禁止自动关闭

bit 3-2 **PSSAC1:PSSAC0:** 引脚 A 和 C 的关闭状态控制位
 1x = 引脚 A 和 C 为三态 (40/44 引脚器件);
 PWM 输出为三态 (28 引脚器件)
 01 = 将引脚 A 和 C 驱动为 1
 00 = 将引脚 A 和 C 驱动为 0

bit 1-0 **PSSBD1:PSSBD0:** 引脚 B 和 D 的关闭状态控制位 ⁽¹⁾
 1x = 引脚 B 和 D 为三态
 01 = 将引脚 B 和 D 驱动为 1
 00 = 将引脚 B 和 D 驱动为 0

注 1: 在 28 引脚器件中, 这些位是保留的, 它们始终保持清零。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

14.4.7.1 自动关闭和自动重启

可以将自动关闭功能配置为在关闭事件后允许模块的自动重启。通过将 ECCP1DEL 寄存器中的 PRSEN 位 (ECCP1DEL<7>) 置 1 使能此功能。

在 PRSEN = 1 (图 14-10) 的关闭模式下, 只要关闭源继续存在, ECCPASE 位将保持置 1。当关闭条件清除时, ECCPASE 位清零。如果 PRSEN = 0 (图 14-11), 一旦出现关闭条件, ECCPASE 位将保持置 1 直到它被固件清零为止。一旦 ECCPASE 被清零, 增强型 PWM 将在下一个 PWM 周期的开始恢复正常操作。

注: 当关闭条件有效时, 禁止写 ECCPASE 位。

如果自动关闭源是某个比较器, 则关闭条件是电平, 与 PRSEN 位的设置无关。只要关闭源继续存在, ECCPASE 位就不能被清零。

通过将 1 写入 ECCPASE 位可强制进入自动关闭模式。

14.4.8 启动注意事项

当在 PWM 模式下使用 ECCP 模块时, 设计硬件时必须要在 PWM 输出引脚上使用适当的外部上拉和 / 或下拉电阻。当单片机复位结束时, 所有的 I/O 引脚都处于高阻态。外部电路必须将电源开关元件保持在关断状态, 直到单片机用适当的信号电平驱动 I/O 引脚, 或激活 PWM 输出为止。

CCP1M1:CCP1M0 位 (CCP1CON<1:0>) 允许用户为每对 PWM 输出引脚 (P1A/P1C 和 P1B/P1D) 选择 PWM 输出信号是高电平有效还是低电平有效。必须在 PWM 引脚被配置为输出之前选择 PWM 输出的极性。建议不要在 PWM 引脚被配置为输出时更改极性配置, 因为这可能会损坏应用电路。

当初始化 PWM 模块时, P1A、P1B、P1C 和 P1D 的输出锁存器可能处于不正确的状态。使能 ECCP 模块同时将 PWM 引脚使能为输出可能导致损坏应用电路。必须先将 ECCP 模块使能为正确的输出模式, 并且必须经过一个完整的 PWM 周期后, 才能将 PWM 引脚配置为输出。当第 2 个 PWM 周期开始时 TMR2IF 位会置 1, 从而表明经过了一个完整的 PWM 周期。

图 14-10: PWM 自动关闭 (PRSEN = 1, 使能自动重启)

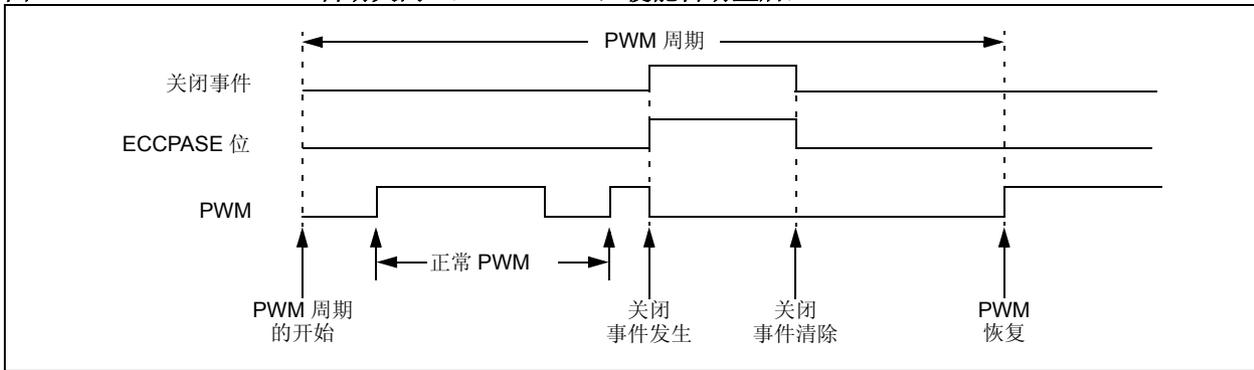
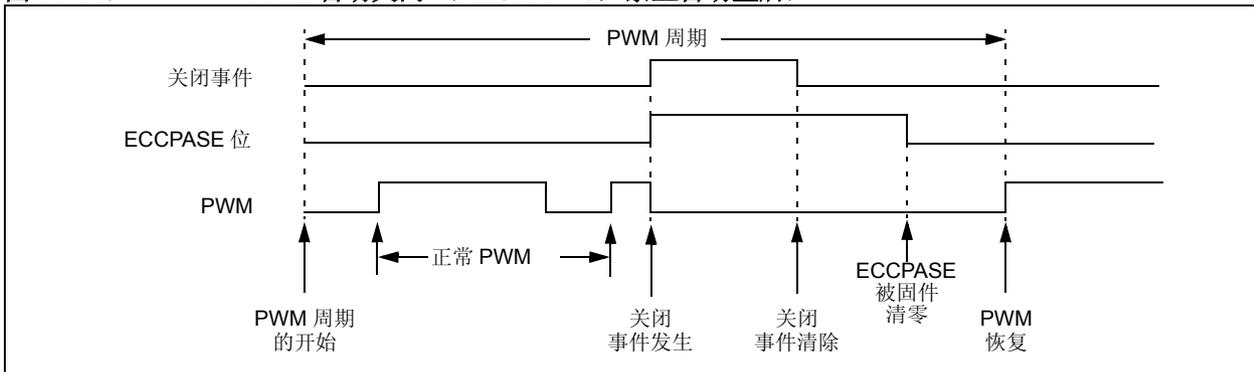


图 14-11: PWM 自动关闭 (PRSEN = 0, 禁止自动重启)



14.4.9 设置 PWM 操作

在将 ECCP 模块配置为 PWM 操作模式时应该执行以下步骤：

1. 通过将相应的 TRIS 位置 1，将 PWM 引脚 P1A 和 P1B（若使用 P1C 和 P1D，还包括这两个引脚）配置为输入。
2. 通过装载 PR2 寄存器设置 PWM 周期。
3. 如果需要设置自动关闭功能：
 - 禁止自动关闭（ECCPASE = 0）
 - 配置自动关闭源（FLT0、比较器 1 或比较器 2）
 - 等待非关闭条件
4. 将 ECCP 模块配置为所需的 PWM 模式，通过用相应的值装载 CCP1CON 寄存器来完成该配置：
 - 通过 P1M1:P1M0 位选择输出配置和方向。
 - 通过 CCP1M3:CCP1M0 位选择 PWM 输出信号的极性。
5. 可通过装载 CCPR1L 寄存器和 CCP1CON<5:4> 位来设置 PWM 占空比。
6. 对于半桥输出模式，通过用合适的值装载 ECCP1DEL<6:0> 来设置死区延迟时间。
7. 如果需要自动关闭操作，则装载 ECCP1AS 寄存器：
 - 使用 ECCPAS2:ECCPAS0 位选择自动关闭源。
 - 使用 PSSAC1:PSSAC0 和 PSSBD1:PSSBD0 位选择 PWM 输出引脚的关闭状态。
 - 将 ECCPASE 位（ECCP1AS<7>）置 1。
 - 使用 CMCON 寄存器配置比较器。
 - 将比较器的输入端配置为模拟输入。
8. 如果需要自动重启操作，则将 PRSEN 位（ECCP1DEL<7>）置 1。
9. 配置并启动 TMR2：
 - 通过清零 TMR2IF（PIR1<1>）中断标志位来清零 TMR2。
 - 通过装载 T2CKPS 位（T2CON<1:0>）来设置 TMR2 预分频值。
 - 通过将 TMR2ON 位（T2CON<2>）置 1 来使能 Timer2。
10. 在新的 PWM 周期开始后，使能 PWM 输出：
 - 等待 TMRn 溢出（TMRnIF 位置 1）。
 - 通过分别清零相应的 TRIS 位，将 CCP1/P1A、P1B、P1C 和 / 或 P1D 引脚使能为输出。
 - 将 ECCPASE 位（ECCP1AS<7>）清零。

14.4.10 功耗管理模式下的操作

在休眠模式下，禁止所有时钟源。Timer2 将不会递增并且模块的状态将保持不变。如果 CCP1 引脚有输出，在休眠模式下将继续保持该输出值不变。当器件被唤醒时，它将继续工作。如果使能了双速启动，INTOSC 和后分频器的起振频率在开始时可能不会太稳定。

在 PRI_IDLE 模式下，主时钟将继续为 ECCP 模块提供时钟。在其他功耗管理模式下，将使用选定的功耗管理模式时钟作为 Timer2 的时钟源。其他功耗管理模式时钟的频率很可能与主时钟频率不同。

14.4.10.1 使用故障保护时钟监视器时的操作

如果使能了故障保护时钟监视器，时钟故障将强制器件进入功耗管理 RC_RUN 模式，并且 OSCFIF 位（PIR2<7>）将置 1。然后将由内部振荡器时钟源为 ECCP 提供时钟，该时钟的频率与主时钟不同。

更多详细信息请查看前面的章节。

14.4.11 复位的影响

上电复位和后续的复位会将所有端口强制为输入模式，并强制 CCP 寄存器进入它们的复位状态。

这会强制增强型 CCP 模块进入与标准 CCP 模块兼容的状态。

PIC18F45J10 系列

表 14-3: 与 ECCP1 模块和 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
RCON	IPEN	—	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}	42
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
TRISB	PORTB 数据方向控制寄存器								46
TRISC	PORTC 数据方向控制寄存器								46
TRISD ⁽¹⁾	PORTD 数据方向控制寄存器								46
TMR1L	Timer1 寄存器的低字节								44
TMR1H	Timer1 寄存器的高字节								44
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	44
TMR2	Timer2 寄存器								44
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	44
PR2	Timer2 周期寄存器								44
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								45
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								45
CCP1CON	P1M1 ⁽¹⁾	P1M0 ⁽¹⁾	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	45
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 ⁽¹⁾	PSSBD0 ⁽¹⁾	45
ECCP1DEL	PRSEN	PDC6 ⁽¹⁾	PDC5 ⁽¹⁾	PDC4 ⁽¹⁾	PDC3 ⁽¹⁾	PDC2 ⁽¹⁾	PDC1 ⁽¹⁾	PDC0 ⁽¹⁾	45

图注: — = 未用 (读为 0)。在 ECCP 运行期间不使用阴影单元。

注 1: 在 28 引脚器件中这些寄存器和 / 或位不被使用, 并且应该读为 0。

15.0 主控同步串行端口 (MSSP) 模块

15.1 主控 SSP (MSSP) 模块概述

主控同步串行端口 (Master Synchronous Serial Port, MSSP) 模块是用于同其他外设或单片机进行通信的串行接口。这些外设器件可以是串行 EEPROM、移位寄存器、显示驱动器或 A/D 转换器等。MSSP 模块有下列两种工作模式：

- 串行外设接口 (SPI)
- I²C
 - 全主控模式
 - 从动模式 (支持广播地址呼叫)

I²C 接口在硬件上支持下列模式：

- 主控模式
- 多主机模式
- 从动模式

PIC18F24J10/25J10 (28 引脚) 器件具有一个 MSSP 模块, 称为 MSSP1。PIC18F44J10/45J10 (40/44 引脚) 器件具有两个 MSSP 模块, 称为 MSSP1 和 MSSP2。每个模块均可独立工作。

注： 在本章中, 在所有工作模式下, 通常指的 MSSP 模块都可以被认为是 MSSP1 或 MSSP2。如果需要, 寄存器名称和模块 I/O 信号使用通用标识符 “x” 来区分某个特定模块。控制位名称没有区别。

15.2 控制寄存器

每个 MSSP 模块有三个相关的控制寄存器, 包括一个状态寄存器 (SSPxSTAT) 和两个控制寄存器 (SSPxCON1 和 SSPxCON2)。根据 MSSP 模块是在 SPI 模式还是 I²C 模式下工作, 这些寄存器的用途及各自的配置位将大相径庭。

下面各章节将提供更多细节。

注： 在具有多个 MSSP 模块的器件中, 要特别注意 SSPCON 寄存器的名称。SSP1CON1 和 SSP1CON2 控制同一模块工作的不同方面, 而 SSP1CON1 和 SSP2CON1 控制两个不同模块的相同功能。

15.3 SPI 模式

SPI 模式允许同步发送和接收 8 位数据。该系列器件支持 SPI 的所有四种模式。通常使用以下三个引脚来完成通信：

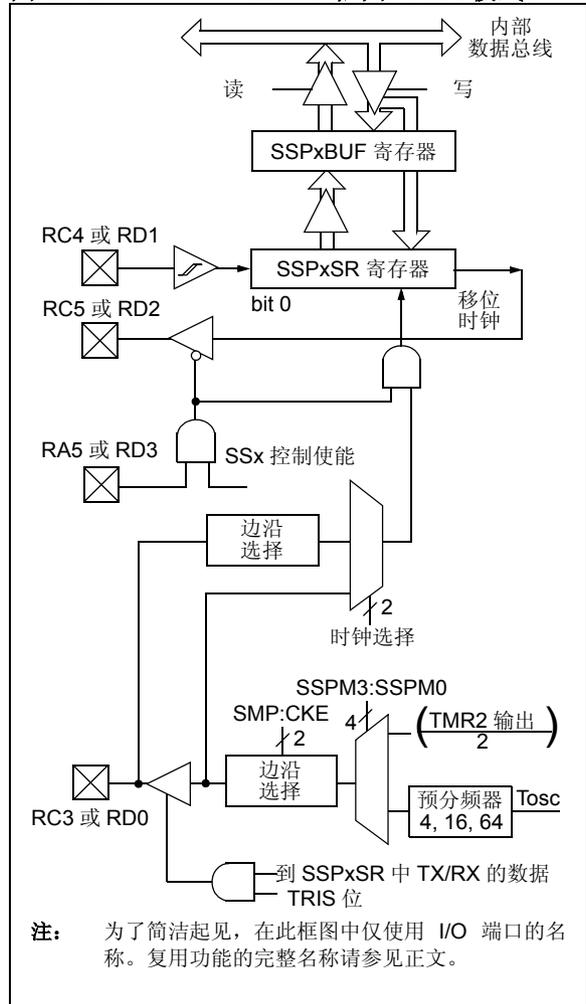
- 串行数据输出 (SDOx) – RC5/SDO1 或 RD2/PSP2/SDO2
- 串行数据输入 (SDIx) – RC4/SDI1/SDA1 或 RD1/PSP1/SDI2/SDA2
- 串行时钟 (SCKx) – RC3/SCK1/SCL1 或 RD0/PSP0/SCK2/SCL2

此外, 当工作在从动模式下时, 会用到第 4 根引脚：

- 从动选择 (\overline{SSx}) – RA5/AN4/ $\overline{SS1}$ /C2OUT 或 RD3/PSP3/ $\overline{SS2}$

图 15-1 给出了 MSSP 模块在 SPI 模式下的框图。

图 15-1: MSSP 框图 (SPI 模式)



PIC18F45J10 系列

15.3.1 寄存器

每个MSSP模块都有四个用于控制其在SPI模式下工作的寄存器。它们是：

- MSSP 控制寄存器 1 (SSPxCON1)
- MSSP 状态寄存器 (SSPxSTAT)
- 串行接收 / 发送缓冲寄存器 (SSPxBUF)
- MSSP 移位寄存器 (SSPxSR) ——不能直接访问

SSPxCON1 和 SSPxSTAT 是在 SPI 模式下工作的控制寄存器和状态寄存器。SSPxCON1 寄存器是可读写的。SSPxSTAT 的低六位是只读的，其高两位是可读写的。

SSPxSR 是用来将数据移入或移出的移位寄存器。SSPxBUF是缓冲寄存器，可用于缓存读写的数据字节。

接收时，SSPxSR 和 SSPxBUF 共同构成一个双缓冲接收器。当 SSPxSR 接收到一个完整的字节后，该字节被送入 SSPxBUF 寄存器，同时中断标志位 SSPxIF 置 1。

在发送过程中，SSPxBUF 并不是双重缓冲的。对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPxSR。

寄存器 15-1: SSPxSTAT: MSSPx 状态寄存器 (SPI 模式)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7						bit 0	

bit 7 **SMP:** 采样位

SPI 主控模式:

1 = 在数据输出时间的末尾采样输入数据

0 = 在数据输出时间的中间采样输入数据

SPI 从动模式:

当使用 SPI 从动模式时，必须将 SMP 清零。

bit 6 **CKE:** SPI 时钟选择位

1 = 当时钟从有效状态变为空闲状态时开始发送

0 = 当时钟从空闲状态变为有效状态时开始发送

注: 时钟状态的极性由 CKP 位 (SSPxCON1<4>) 设置。

bit 5 **D/A:** 数据 / 地址位

仅在 I²C 模式下使用。

bit 4 **P:** 停止位

仅在 I²C 模式下使用。当 MSSP 模块被禁止 (SSPEN 清零) 时该位被清零。

bit 3 **S:** 启动位

仅在 I²C 模式下使用。

bit 2 **R/W:** 读 / 写信息位

仅在 I²C 模式下使用。

bit 1 **UA:** 更新地址位

仅在 I²C 模式下使用。

bit 0 **BF:** 缓冲器满状态位 (仅用于接收模式)

1 = 接收完成, SSPxBUF 已满

0 = 接收未完成, SSPxBUF 为空

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 15-2: SSPxCON1: MSSPx 控制寄存器 1 (SPI 模式)

R/W-0								
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	
bit 7								bit 0

- bit 7 **WCOL:** 写冲突检测位 (仅用于发送模式)
 1 = 当仍然在发送前一个字时发生了对 SSPxBUF 寄存器的写操作 (必须由软件清零)
 0 = 未发生冲突
- bit 6 **SSPOV:** 接收溢出标志位
SPI 从动模式:
 1 = SSPxBUF 寄存器仍在保存前一数据时, 接收到一个新的字节。一旦发生溢出, SSPxSR 中的数据就会丢失。溢出只发生在从动模式下。即使只是发送数据, 用户也必须读 SSPxBUF, 以避免将溢出标志位置 1 (必须由软件清零)。
 0 = 未发生溢出
注: 在主控模式下, 溢出位不会被置 1, 因为每次接收 (和发送) 新数据, 都要从写入 SSPxBUF 寄存器开始。
- bit 5 **SSPEN:** 主控同步串行端口使能位
 1 = 使能串行端口并将 SCKx、SDOx、SDIx 和 \overline{SSx} 配置为串行端口引脚
 0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚
注: 当该位使能时, 必须将这些引脚正确配置为输入或输出。
- bit 4 **CKP:** 时钟极性选择位
 1 = 时钟高电平为空闲状态
 0 = 时钟低电平为空闲状态
- bit 3-0 **SSPM3:SSPM0:** 主控同步串行端口模式选择位
 0101 = SPI 从动模式, 时钟 = SCKx 引脚, 禁止 \overline{SSx} 引脚控制, 可将 \overline{SSx} 用作 I/O 引脚
 0100 = SPI 从动模式, 时钟 = SCKx 引脚, 使能 \overline{SSx} 引脚控制功能
 0011 = SPI 主控模式, 时钟 = TMR2 输出 /2
 0010 = SPI 主控模式, 时钟 = Fosc/64
 0001 = SPI 主控模式, 时钟 = Fosc/16
 0000 = SPI 主控模式, 时钟 = Fosc/4
注: 这里未列出的位组合为保留的或只在 I²C 模式中使用。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

PIC18F45J10 系列

15.3.2 工作原理

当初始化 SPI 时，需要指定几个选项。可以通过对相应的控制位（SSPxCON1<5:0> 和 SSPxSTAT<7:6>）编程来指定。这些控制位用于指定以下选项：

- 主控模式（SCKx 作为时钟输出）
- 从动模式（SCKx 作为时钟输入）
- 时钟极性（SCKx 的空闲状态）
- 输入数据的采样相位（数据输出时间的中间或末尾）
- 时钟边沿（在 SCKx 上升沿 / 下降沿输出数据）
- 时钟速率（仅用于主控模式）
- 从动选择模式（仅用于从动模式）

MSSP 模块由一个发送 / 接收移位寄存器（SSPxSR）和一个缓冲寄存器（SSPxBUF）组成。SSPxSR 将数据移入和移出器件，最高有效位在前。SSPxBUF 保存上次写入 SSPxSR 的数据直到新接收到的数据就绪为止。一旦 8 位数据接收完毕，该字节就被移入 SSPxBUF 寄存器。然后，缓冲器满检测位 BF（SSPxSTAT<0>）和中断标志位 SSPxIF 被置 1。这种双重缓冲数据接收方式

（SSPxBUF）允许在读取刚接收的数据之前，就开始接收下一个字节。在数据发送 / 接收期间，任何试图写 SSPxBUF 寄存器的操作都会被忽略，并将写冲突检测位 WCOL（SSPxCON1<7>）置 1。此时用户必须用软件将 WCOL 位清零，否则无法判别下一次对 SSPxBUF 的写操作是否成功完成。

应用软件要想接收到有效数据，应该在下一个要传送的数据字节写入 SSPxBUF 之前，将 SSPxBUF 中的前一个数据读出。缓冲器满标志位 BF（SSPxSTAT<0>）用于表示何时 SSPxBUF 已经载入了接收到的数据（发送完成）。当 SSPxBUF 中的数据被读取后，BF 位即被清零。如果 SPI 仅作为一个发送器，则不必理会接收的数据。通常可用 MSSP 中断来判断发送或接收何时完成。必须读和 / 或写 SSPxBUF。如果不使用中断来处理数据的收发，用软件查询方法同样可确保不会发生写冲突。例 15-1 显示了装载 SSP1BUF（SSP1SR）进行数据发送的过程。

不能直接读写 SSPxSR 寄存器，只能通过访问 SSPxBUF 寄存器才能访问它。此外，SSPxSTAT 寄存器表示各种状态。

例 15-1: 装载 SSP1BUF（SSP1SR）寄存器

LOOP	BTFSS	SSP1STAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSP1BUF, W	;WREG reg = contents of SSP1BUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSP1BUF	;New data to xmit

15.3.3 使能 SPI I/O

要使能串行端口，MSSP使能位SSPEN (SSPxCON1<5>) 必须置1。要复位或重新配置 SPI 模式，要先将 SSPEN 位清零，重新初始化 SSPxCON 寄存器，然后将 SSPEN 位置1。这将把 SDIx、SDOx、SCKx 和 SSx 引脚配置为串行端口引脚。要将这些引脚用作串行端口，还必须将其数据方向位（在 TRIS 寄存器中）正确编程，方法如下：

- SDIx 由 SPI 模块自动控制
- SDOx 必须将 TRISC<5>（或 TRISD<2>）位清零
- SCKx（主控模式）必须将 TRISC<3>（或 TRISD<0>）位清零
- SCKx（从动模式）必须将 TRISC<3>（或 TRISD<0>）位置1
- SSx 必须将 TRISA<5>（或 TRISD<3>）位置1

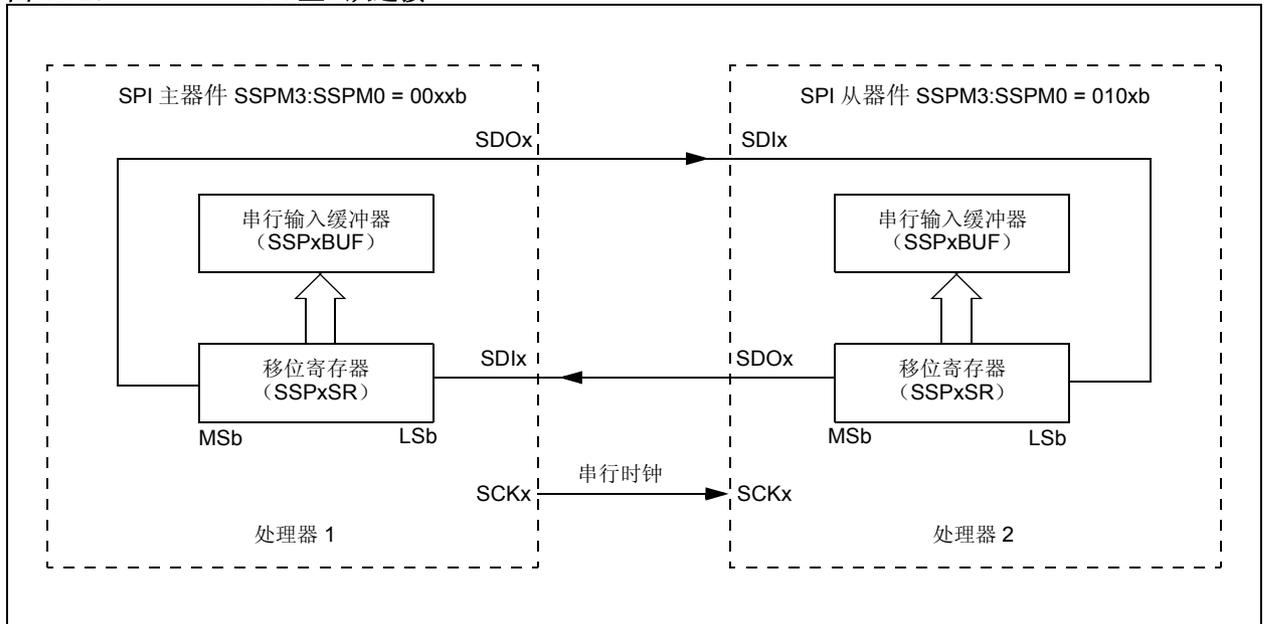
对于不需要的任何串行端口功能，可通过将对应的数据方向（TRIS）寄存器设置为相反值来改写。

15.3.4 典型连接

图 15-2所示为两个单片机之间的典型连接。主器件（处理器 1）通过发送 SCKx 信号来启动数据传输。数据在编程设定的时钟边沿被移出两个处理器的移位寄存器，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（CKP）设定为相同，这样两个控制器就可以同时收发数据。数据是否有效取决于应用软件。这就导致以下三种数据传输的情况：

- 主器件发送数据 – 从器件发送无效（Dummy）数据
- 主器件发送数据 – 从器件发送数据
- 主器件发送无效数据 – 从器件发送数据

图 15-2: SPI 主/从连接



PIC18F45J10 系列

15.3.5 主控模式

主器件可以随时启动数据传输，因为它控制 SCKx。主器件根据软件协议确定从器件（处理器 2，图 15-2）应在何时广播数据。

在主控模式下，数据一旦写入 SSPxBUF 寄存器就开始发送或接收。如果 SPI 仅作为接收器，则可以禁止 SDOx 输出（将其编程设定为输入）。SSPxSR 寄存器按设置的时钟速率，对 SDIx 引脚上的信号进行连续移位输入。每个字节接收完后，会被装入 SSPxBUF 寄存器，就像普通的接收字节一样（相应的中断和状态位置 1）。这在以“线路活动监控”（Line Activity Monitor）方式工作的接收器中很有用。

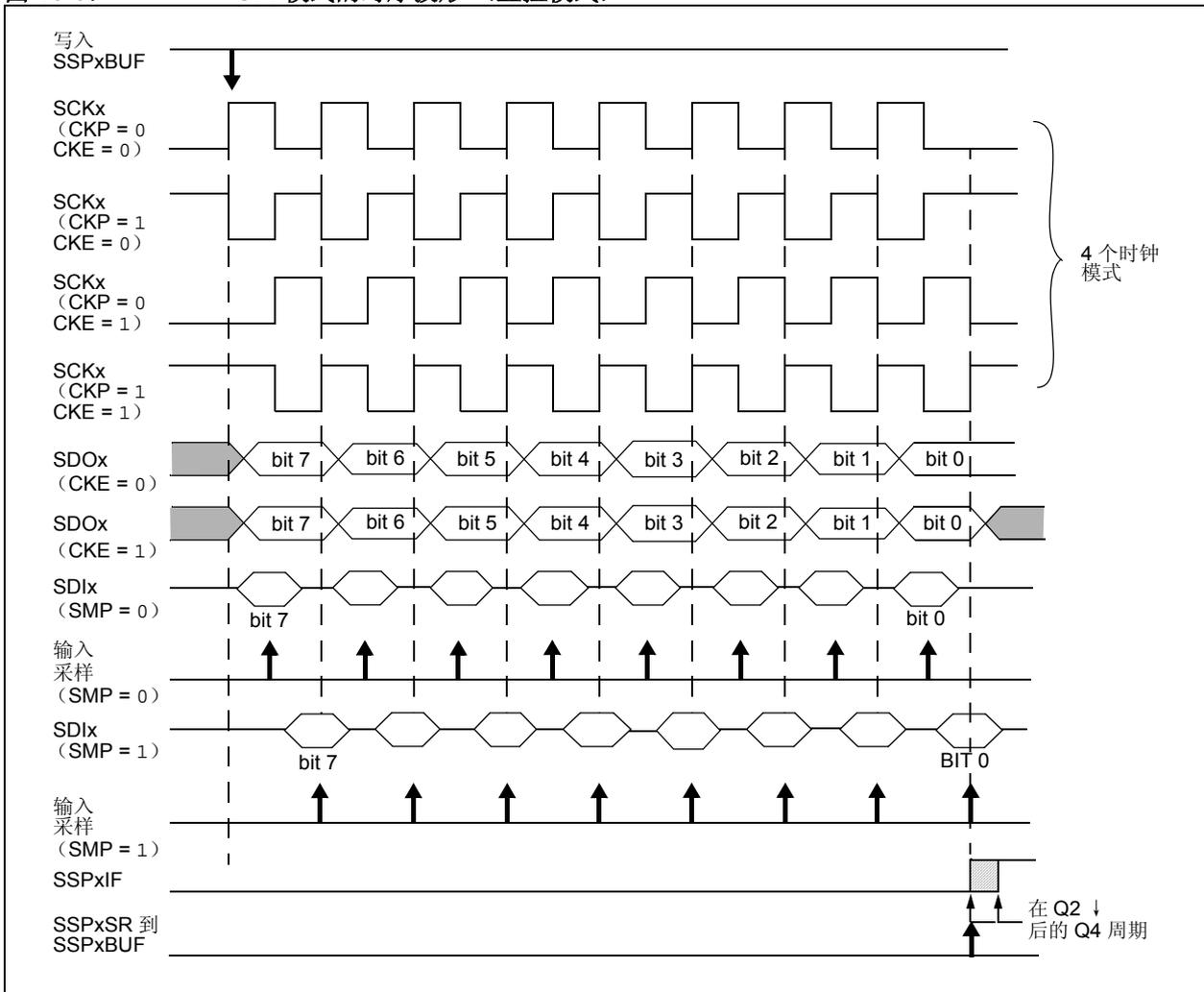
可通过对 CKP 位（SSPxCON1<4>）进行相应的编程来选择时钟极性。图 15-3、图 15-5 和图 15-6 将给出 SPI 通信波形图，其中首先发送的是 MSB。在主控模式下，SPI 时钟速率（比特率）可由用户编程设定为下列速率之一：

- Fosc/4（或 Tcy）
- Fosc/16（或 4•Tcy）
- Fosc/64（或 16•Tcy）
- Timer2 输出 /2

这里允许的最大数据速率是 10.00 Mbps（当晶振为 40 MHz 时）。

图 15-3 给出了主控模式的波形图。当 CKE 位置 1 时，SDOx 数据在 SCKx 上出现时钟边沿前就有效。图中所示输入采样的变化由 SMP 位的状态决定。图中指出了将接收到的数据装入 SSPxBUF 的时间。

图 15-3: SPI 模式的时序波形（主控模式）



15.3.6 从动模式

在从动模式下，当 SCKx 引脚上出现外部时钟脉冲时，发送和接收数据。当最后一位数据被锁存后，中断标志位 SSPxIF 置 1。

在 SPI 从动模式下使能模块之前，时钟线必须处于适当的空闲状态。可以通过读 SCKx 引脚来观察时钟线。由 CKP 位 (SSPxCON1<4>) 确定空闲状态。

在从动模式下，时钟由 SCKx 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠状态下，从器件仍可发送 / 接收数据。当收到一个字节时，器件从休眠状态被唤醒。

15.3.7 从动选择同步

SSx 引脚允许器件工作在同步从动模式。SPI 必须在从动模式下，并使能 SSx 引脚控制 (SSPxCON1<3:0>=04h)。当 SSx 引脚为低电平时，

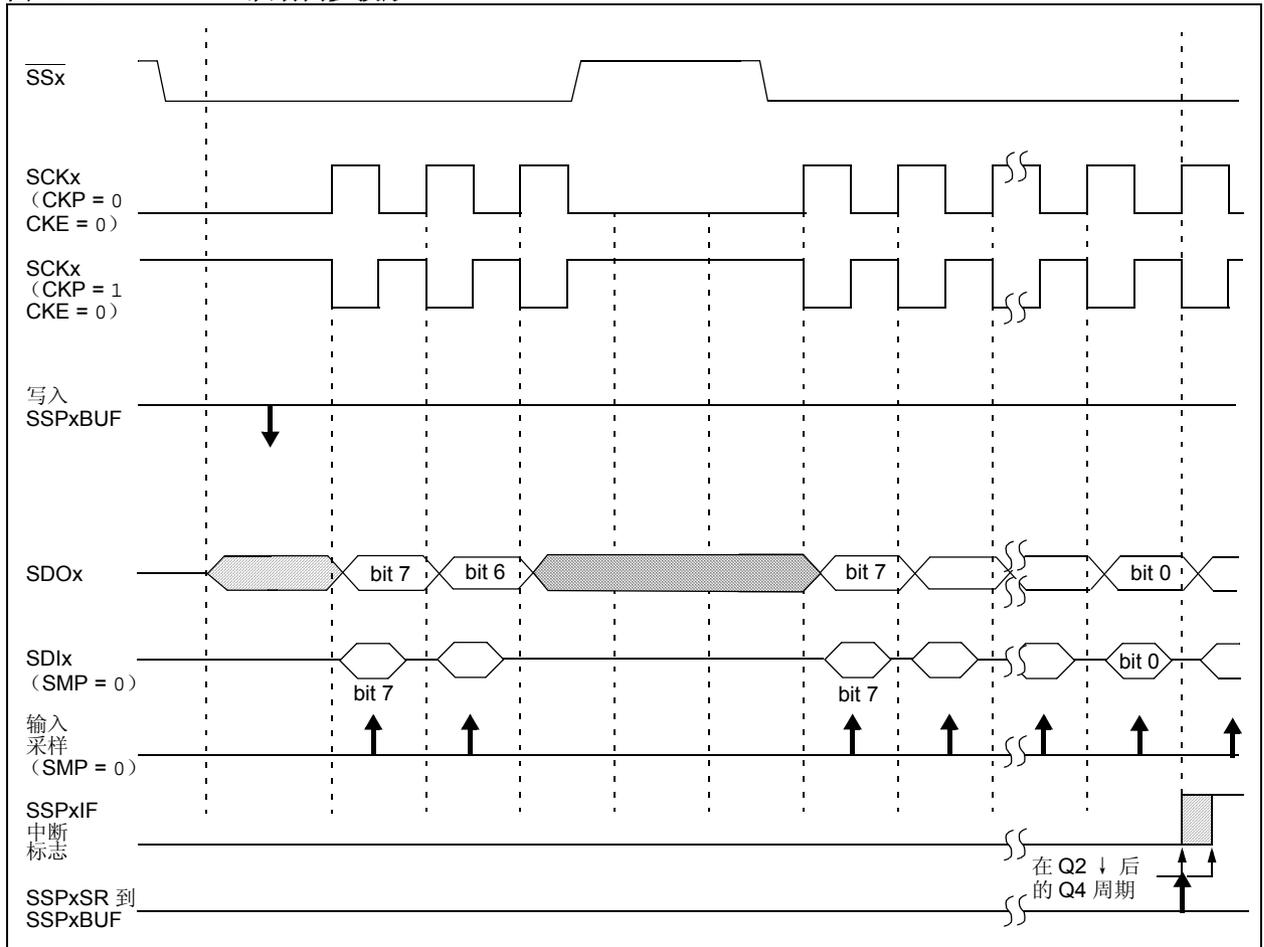
使能数据的发送和接收，同时 SDOx 引脚被驱动。当 SSx 引脚为高电平时，即使是在数据的发送过程中，SDOx 引脚也不再被驱动，而是变成悬空输出。根据应用的需要，可外接上拉 / 下拉电阻。

- 注 1:** 当 SPI 工作在从动模式下，并且 SSx 引脚控制使能 (SSPxCON1<3:0>=0100) 时，如果 SSx 引脚置为 VDD 电平将使 SPI 模块复位。
- 2:** 如果 SPI 工作在从动模式下并且 CKE 置 1，则必须使能 SSx 引脚控制。

当 SPI 模块复位后，位计数器被强制归 0。这可以通过强制将 SSx 引脚拉为高电平或将 SSPEN 位清零来实现。

将 SDOx 引脚和 SDIx 引脚相连，可以仿真双线制通信。当 SPI 需要作为接收器工作时，SDOx 引脚可以被配置为输入。这样就禁止了从 SDOx 发送数据。因为 SDIx 不会引起总线冲突，因而总是可以将其保留为输入 (SDIx 功能)。

图 15-4: 从动同步波形



PIC18F45J10 系列

图 15-5: SPI 模式时序波形 (从动模式, CKE = 0)

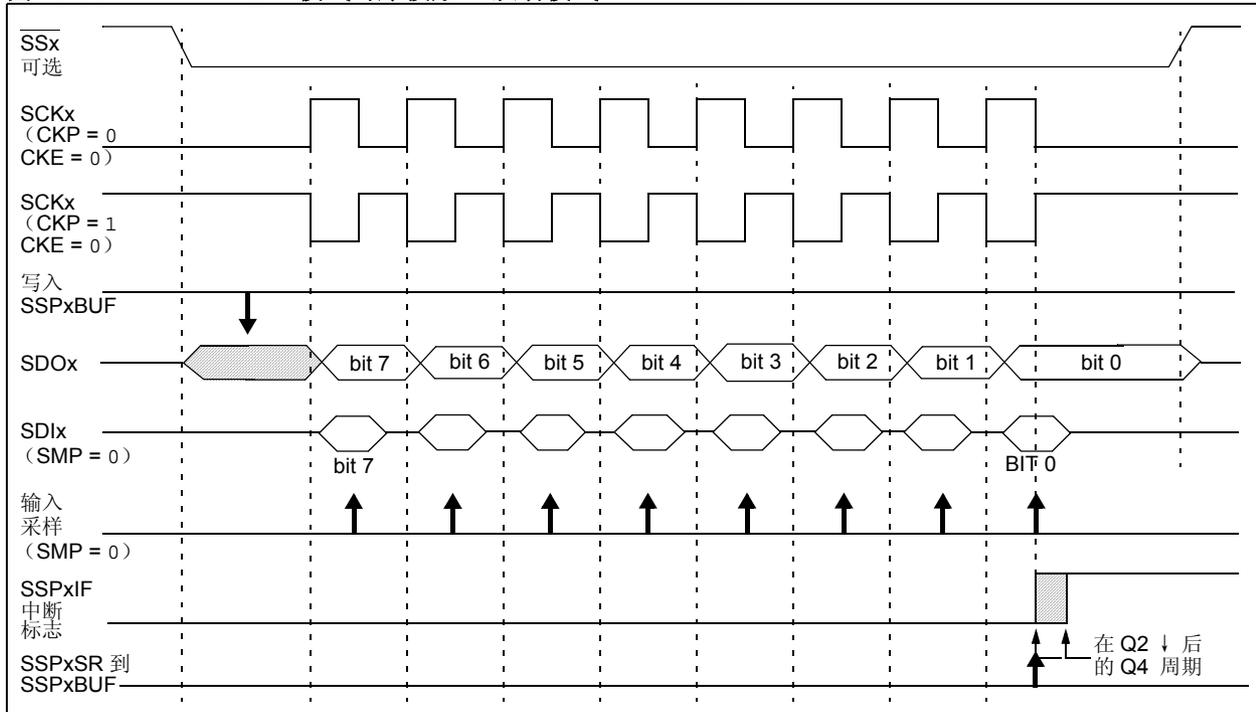
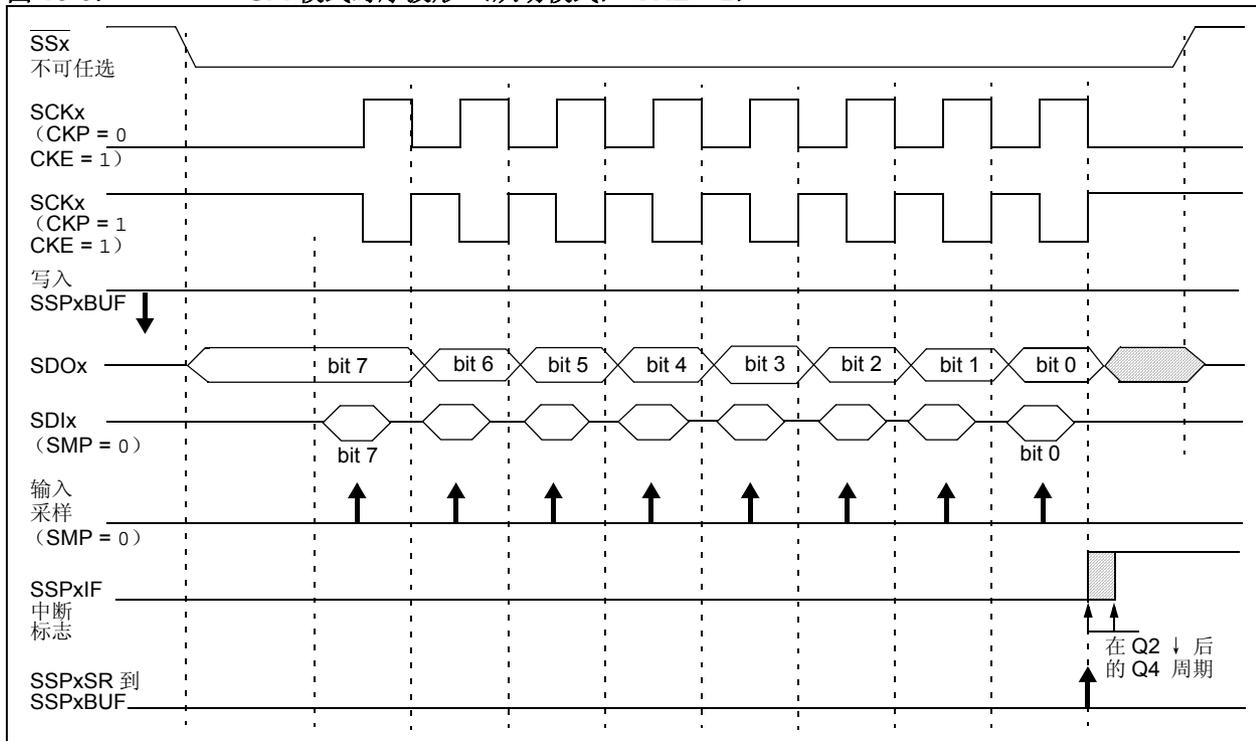


图 15-6: SPI 模式时序波形 (从动模式, CKE = 1)



15.3.8 在功耗管理模式下的操作

在 SPI 主控模式下，模块时钟速度可以与全功耗模式下的不同；处于休眠模式时，所有时钟都停止。

在空闲模式下，需要为外设提供一个时钟。此时钟应该来自于主时钟源、辅助时钟源（32.768 kHz 的 Timer1 振荡器）或 INTOSC 时钟源。更多信息，请参见第 2.6 节“时钟源和振荡器切换”。

在大多数情况下，主器件为 SPI 数据提供的时钟速率并不重要；但是，每个系统都应该评估此因素。

如果使能了 MSSP 中断，那么当主器件发送完数据时这些中断可以将控制器从休眠模式或某种空闲模式唤醒。如果不想从休眠或空闲模式退出，应该禁止 MSSP 中断。

如果选择了休眠模式，所有模块的时钟都将停止，并且在器件被唤醒前，发送 / 接收将保持此停滞状态。当器件返回到运行模式后，该模块将恢复发送和接收数据。

在 SPI 从动模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件处于任何功耗管理模式下，而且数据仍可被移入 SPI 发送 / 接收移位寄存器。当 8 位数据全部接收到后，MSSP 中断标志位将置 1，并且如果使能中断的话，将唤醒器件。

15.3.9 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

15.3.10 总线模式兼容性

表 15-1 给出了标准 SPI 模式和 CKP 与 CKE 控制位状态的对应情况。

表 15-1: SPI 总线模式

标准 SPI 模式术语	控制位状态	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

还有一个 SMP 位用来控制何时采样数据。

15.3.11 SPI 时钟速度和模块相互关系

因为 MSSP1 和 MSSP2 是独立的模块，它们可以不同的数据速率同时工作。设置 SSPxCON1 寄存器的 SSPM3:SSPM0 位的值将确定相应模块的速率。

有一种例外情况就是在两个模块都在主控模式下使用 Timer2 作为时基的时候。在这种情况下，任何对 Timer2 工作模式的更改都会对两个 MSSP 模块造成相同的影响。如果每个模块需要不同的比特率，用户应该为其中一个模块选择其他三种时基中的一种。

PIC18F45J10 系列

表 15-2: 与 SPI 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
PIR3	SSP2IF	BCL2IF	—	—	—	—	—	—	45
PIE3	SSP2IE	BCL2IE	—	—	—	—	—	—	45
IPR3	SSP2IP	BCL2IP	—	—	—	—	—	—	45
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	46
TRISD ⁽¹⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	46
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								44
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	44
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	44
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								46
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	46
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	46

图注: SPI 模式下的 MSSP 模块不使用阴影单元。

注 1: 在 28 引脚器件中这些寄存器和 / 或位不被使用, 并且应该读为 0。

15.4 I²C 模式

MSSP 模块工作在 I²C 模式时，可以实现所有的主控和从动功能（包括广播呼叫支持），并且用硬件提供启动位和停止位的中断来判断总线何时空闲（多主机功能）。MSSP 模块实现了标准模式规范，以及 7 位和 10 位寻址。

有两个引脚用于数据传输：

- 串行时钟 (SCLx) —— RC3/SCK1/SCL1 或 RD0/PSP0/SCK2/SCL2
- 串行数据 (SDAx) —— RC4/SDI1/SDA1 或 RD1/PSP1/SDI2/SDA2

用户必须通过置位 TRISC<4:3> 或 TRISD<1:0> 位将这些引脚配置为输入引脚。

15.4.1 寄存器

MSSP 模块有 6 个寄存器与 I²C 操作相关。它们是：

- MSSP 控制寄存器 1 (SSPxCON1)
- MSSP 控制寄存器 2 (SSPxCON2)
- MSSP 状态寄存器 (SSPxSTAT)
- 串行接收 / 发送缓冲寄存器 (SSPxBUF)
- MSSP 移位寄存器 (SSPxSR)：不能直接访问
- MSSP 地址寄存器 (SSPxADD)

SSPxCON1、SSPxCON2 和 SSPxSTAT 是在 I²C 模式下工作的控制寄存器和状态寄存器。SSPxCON1 和 SSPxCON2 寄存器是可读写的。SSPxSTAT 的低六位是只读的，其高两位是可读写的。

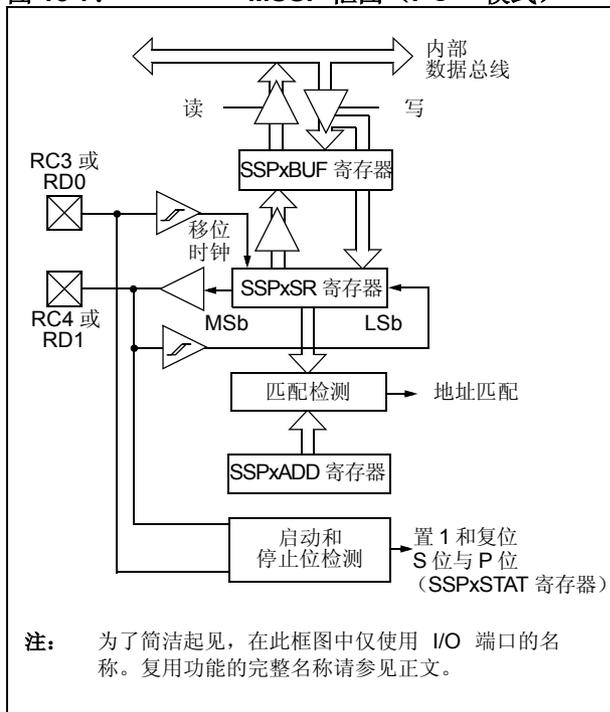
SSPxSR 是用来将数据移入或移出的移位寄存器。SSPxBUF 是缓冲寄存器，可用于缓存读写的数据字节。

将 MSSP 配置为工作于 I²C 从动模式时，SSPxADD 寄存器将保存从器件地址。当将 MSSP 配置为工作于主控模式时，SSPxADD 的低 7 位用作波特率发生器的重载值。

接收时，SSPxSR 和 SSPxBUF 共同构成一个双缓冲接收器。当 SSPxSR 接收到一个完整的字节后，该字节被送入 SSPxBUF 寄存器，同时中断标志位 SSPxIF 置 1。

在发送过程中，SSPxBUF 并不是双重缓冲的。对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPxSR。

图 15-7: MSSP 框图 (I²C™ 模式)



PIC18F45J10 系列

寄存器 15-3: SSPxSTAT: MSSPx 状态寄存器 (I²C™ 模式)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF

bit 7

bit 0

bit 7 **SMP:** 变化率控制位

在主控或从动模式下:

1 = 禁止变化率控制以适应标准速度模式 (100 kHz 和 1 MHz)

0 = 使能变化率控制以适应高速模式 (400kHz)

bit 6 **CKE:** SMBus 选择位

在主控或从动模式下:

1 = 使能 SMBus 特定输入

0 = 禁止 SMBus 特定输入

bit 5 **D/A:** 数据 / 地址位

在主控模式下:

保留。

在从动模式下:

1 = 表示上一个接收或发送的字节是数据

0 = 表示上一个接收或发送的字节是地址

bit 4 **P:** 停止位

1 = 表示上一次检测到了停止位

0 = 表示上一次未检测到停止位

注: 当复位或 SSPEN 被清零时该位清零。

bit 3 **S:** 起始位

1 = 表示上一次检测到了起始位

0 = 表示上一次未检测到起始位

注: 当复位或 SSPEN 被清零时该位清零。

bit 2 **R/W:** 读 / 写信息位 (仅用于 I²C 模式)

在从动模式下:

1 = 读

0 = 写

注: 该位用来保存在最近一次地址匹配后的 $\overline{R/W}$ 位信息。该位仅在从地址匹配开始到下一个起始位、停止位或非 ACK 位被检测到期间有效。

在主控模式下:

1 = 正在进行发送

0 = 未进行发送

注: 该位与 SEN、RSEN、PEN、RCEN 或 ACKEN 进行“或”运算的结果表示 MSSP 是否处于活动模式。

bit 1 **UA:** 更新地址位 (仅用于 10 位从动模式)

1 = 表示用户需要更新 SSPxADD 寄存器中的地址

0 = 不需要更新地址

bit 0 **BF:** 缓冲器满状态位

在发送模式下:

1 = SSPxBUF 已满

0 = SSPxBUF 为空

在接收模式下:

1 = SSPxBUF 已满 (不包括 \overline{ACK} 位和停止位)

0 = SSPxBUF 为空 (不包括 \overline{ACK} 位和停止位)

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 15-4: SSPxCON1: MSSPx 控制寄存器 1 (I²C™ 模式)

R/W-0							
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** 写冲突检测位

在主控发送模式下:

1 = 当 I²C 信号处于发送数据无效状态时, 试图写 SSPxBUF 寄存器 (必须由软件清零)
0 = 未发生冲突

在从动发送模式下:

1 = 当仍然在发送前一个字时发生了对 SSPxBUF 寄存器的写操作 (必须由软件清零)
0 = 未发生冲突

在接收模式下 (主控或从动模式):

可为任意值。

bit 6 **SSPOV:** 接收溢出标志位

在接收模式下:

1 = SSPxBUF 寄存器仍在保存前一数据时, 接收到一个新的字节 (必须由软件清零)
0 = 未发生溢出

在发送模式下:

可为任意值。

bit 5 **SSPEN:** 主控同步串行端口使能位

1 = 使能串行端口并将 SDAx 和 SCLx 引脚配置为串行端口引脚。
0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚

注: 当该位被使能时, 必须将 SDAx 和 SCLx 引脚正确地配置为输入引脚或输出引脚。

bit 4 **CKP:** SCKx 释放控制位

在从动模式下:

1 = 释放时钟
0 = 保持时钟低电平 (时钟延长), 用来确保数据建立时间

在主控模式下:

在此模式下未使用。

bit 3-0 **SSPM3:SSPM0:** 同步串行端口模式选择位

1111 = I²C 从动模式, 10 位地址, 并使能起始位和停止位中断

1110 = I²C 从动模式, 7 位地址, 并使能起始位和停止位中断

1011 = I²C 由固件控制的主控模式 (从动空闲)

1000 = I²C 主控模式, 时钟 = Fosc/(4 * (SSPxADD + 1))

0111 = I²C 从动模式, 10 位地址

0110 = I²C 从动模式, 7 位地址

这里未列出的位组合为保留的或只在 SPI 模式下使用。

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

寄存器 15-5: SSPxCON2: MSSPx 控制寄存器 2 (I²C™ 模式)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
GCEN	ACKSTAT	ACKDT	ACKEN ⁽¹⁾	RCEN ⁽¹⁾	PEN ⁽¹⁾	RSEN ⁽¹⁾	SEN ⁽¹⁾	
bit 7								bit 0

- bit 7 **GCEN:** 广播呼叫使能位 (仅用于从动模式)
 1 = 当在 SSPxSR 中接收到广播呼叫地址 (0000h) 时允许中断
 0 = 禁止广播呼叫地址
- bit 6 **ACKSTAT:** 应答状态位 (仅用于主控发送模式)
 1 = 未收到来自从器件的应答
 0 = 接收到来自从器件的应答
- bit 5 **ACKDT:** 应答数据位 (仅用于主控接收模式)
 1 = 不应答
 0 = 应答
- 注:** 用户在接收结束时启动一个应答序列, 同时发送该值。
- bit 4 **ACKEN:** 应答序列使能位 (仅用于主控接收模式) ⁽¹⁾
 1 = 在 SDAx 和 SCLx 引脚上启动应答序列, 并发送 ACKDT 数据位。由硬件自动清零。
 0 = 应答序列空闲
- bit 3 **RCEN:** 接收使能位 (仅用于主控模式) ⁽¹⁾
 1 = 使能 I²C 接收模式
 0 = 接收空闲
- bit 2 **PEN:** 停止条件使能位 (仅用于主控模式) ⁽¹⁾
 1 = 在 SDAx 和 SCLx 引脚上启动停止条件。由硬件自动清零。
 0 = 停止条件空闲
- bit 1 **RSEN:** 重复启动条件使能位 (仅用于主控模式) ⁽¹⁾
 1 = 在 SDAx 和 SCLx 引脚上启动重复启动条件。由硬件自动清零。
 0 = 重复启动条件空闲
- bit 0 **SEN:** 启动条件使能 / 延长使能位 ⁽¹⁾
在主控模式下:
 1 = 在 SDAx 和 SCLx 引脚上开始启动条件。由硬件自动清零。
 0 = 启动条件空闲
在从动模式下:
 1 = 为从动发送和从动接收使能时钟延长 (延长已使能)
 0 = 时钟延长被禁止。

注 1: 对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位, 如果 I²C 模块不处于空闲模式, 可能这些位不会被置 1 (没有缓存) 并且也可能不会写入 SSPxBUF (或禁止写 SSPxBUF)。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

15.4.2 工作方式

将 MSSP 使能位 SSPEN (SSPxCON1<5>) 置 1, 可启用 MSSP 模块功能。

SSPxCON1 寄存器可以控制 I²C 操作。可通过设置模式选择位 (SSPxCON1<3:0>) 选择以下几种 I²C 模式之一:

- I²C 主控模式,
时钟 = (Fosc/4) x (SSPxADD + 1)
- I²C 从动模式 (7 位地址)
- I²C 从动模式 (10 位地址)
- I²C 从动模式 (7 位地址), 使能起始位和停止位中断
- I²C 从动模式 (10 位地址), 使能起始位和停止位中断
- I²C 固件控制的主动模式, 从器件空闲

通过置位相应的 TRISC 或 TRISD 位将 SCLx 和 SDAx 引脚编程为输入引脚, 在 SSPEN 位置 1 时选择任一 I²C 模式, 这样将强制 SCLx 和 SDAx 引脚为漏极开路。要确保此模块的正常工作, 必须为 SCLx 和 SDAx 引脚提供外接上拉电阻。

15.4.3 从动模式

在从动模式下, SCLx 引脚和 SDAx 引脚必须被配置为输入 (TRISC<4:3> 或 TRISD<1:0> 置 1)。必要时 (从发送器) MSSP 模块将使用输出数据改写输入状态。

I²C 从动模式硬件会在地址匹配时产生中断。用户也可以通过模式选择位选择使用起始位或停止位中断。

当地址匹配时或在地址匹配后传送的数据被接收时, 硬件会自动产生一个应答 (ACK) 脉冲, 并把当时 SSPxSR 寄存器中接收到的数据装入 SSPxBUF 寄存器。

只要满足下列条件之一, MSSP 模块就不会产生此 ACK 脉冲:

- 缓冲器满标志位 BF (SSPxSTAT<0>) 在接收到传输的数据前置 1。
- 在接收到传输的数据之前, 溢出标志位 SSPOV (SSPxCON1<6>) 已被置 1。

在这种情况下, SSPxSR 寄存器的值不会载入 SSPxBUF, 但是 SSPxIF 位会置 1。BF 位是通过读取 SSPxBUF 寄存器清零的, 而 SSPOV 位是通过软件清零的。

为确保正常工作, SCLx 时钟输入必须满足最小高电平时间和最小低电平时间要求。关于 I²C 规范所规定的高电平和低电平时间以及对 MSSP 模块的具体要求, 请参见时序参数 100 和 101。

15.4.3.1 寻址

一旦使能了 MSSP 模块, 它就会等待启动条件产生。在启动条件出现后, 8 位数据被移入 SSPxSR 寄存器。在时钟 (SCLx) 线的上升沿采样所有的输入位。寄存器 SSPxSR<7:1> 的值会和 SSPxADD 寄存器的值比较, 该比较是在第 8 个时钟脉冲 (SCLx) 的下降沿进行的。如果地址匹配, 并且 BF 位和 SSPOV 位为零, 会发生下列事件:

1. SSPxSR 寄存器的值被装入 SSPxBUF 寄存器。
2. 缓冲器满标志位 BF 置 1。
3. 产生 ACK 脉冲。
4. 在第 9 个 SCLx 脉冲的下降沿, MSSP 中断标志位 SSPxIF 置位 (如果允许中断则产生中断)。

在 10 位地址模式下, 从器件需要接收两个地址字节。第一个地址字节的高 5 位指定这是否为 10 位地址。R/W (SSPxSTAT<2>) 必须指定写操作, 这样从器件才能接收到第二个地址字节。对于 10 位地址, 第一个字节应该是 11110 A9 A8 0, 其中 A9 和 A8 是该地址的两个最高有效位。10 位地址的工作时序如下, 其中 7-9 步是针对从动发送器而言的:

1. 接收地址的第一个 (高) 字节 (SSPxIF 位、BF 位和 UA 位 (SSPxSTAT<1>) 置 1)。
2. 用地址的第二个 (低) 字节刷新 SSPxADD 寄存器 (UA 位清零并释放 SCLx 线)。
3. 读 SSPxBUF 寄存器 (BF 位清零) 并将标志位 SSPxIF 清零。
4. 接收地址的第二个 (低) 字节 (SSPxIF、BF 和 UA 置 1)。
5. 使用地址的第一个 (高) 字节刷新 SSPxADD 寄存器。如果匹配, 释放 SCLx 线, 并将 UA 位清零。
6. 读 SSPxBUF 寄存器 (BF 位清零) 并将标志位 SSPxIF 清零。
7. 重复接收启动条件。
8. 接收地址的第一个 (高) 字节 (SSPxIF 位和 BF 位置 1)。
9. 读 SSPxBUF 寄存器 (BF 位清零) 并将标志位 SSPxIF 清零。

15.4.3.2 接收

当地址字节的 \overline{RW} 位清零并发生地址匹配时，SSPxSTAT 寄存器的 \overline{RW} 位清零。接收的地址被装入 SSPxBUF 寄存器，且 SDAx 数据线保持低电平 (\overline{ACK})。

当存在地址字节溢出条件时，则不会产生应答脉冲 (\overline{ACK})。溢出条件是指 BF 位 (SSPxSTAT<0>) 置 1，或者 SSPOV 位 (SSPxCON1<6>) 置 1。

每个数据传输字节都会产生一个 MSSP 中断。必须用软件将中断标志位 SSPxIF 清零。使用 SSPxSTAT 寄存器可以确定该字节的状态。

如果 SEN 被使能 (SSPxCON2<0> = 1)，SCKx/SCLx (RC3 或 RD0) 将在每个数据传输之后保持为低电平 (时钟延长)。必须通过将 CKP 位 (SSPxCON1<4>) 置 1 才能释放时钟。更多详情，请参见第 15.4.4 节“时钟延长”。

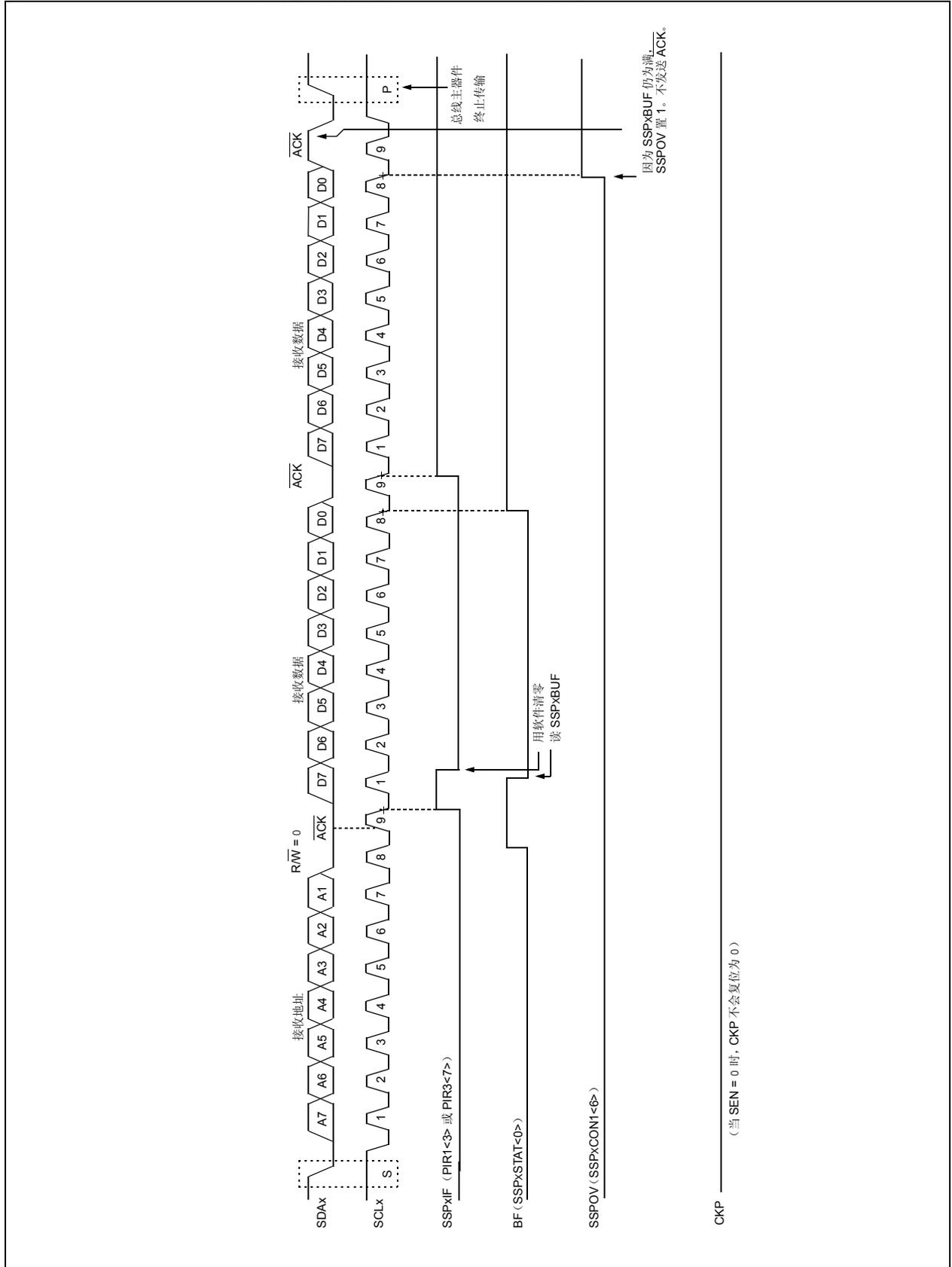
15.4.3.3 发送

当接收的地址字节的 \overline{RW} 位置 1 并发生地址匹配时，SSPxSTAT 寄存器的 \overline{RW} 位置 1。接收到的地址将装入 SSPxBUF 寄存器。不管 SEN 的值如何， \overline{ACK} 脉冲在第 9 位上发送，同时 RC3 或 RD6 引脚保持低电平 (详情请参见第 15.4.4 节“时钟延长”)。通过延长时钟，主器件只有在从器件准备发送数据时才能发出另一个时钟脉冲。传输的数据必须装入 SSPxBUF 寄存器，同时也被装入 SSPxSR 寄存器。然后，应该通过置位 CKP (SSPxCON1<4>) 使能 RC3 或 RD0 引脚。8 个数据位在 SCLx 输入信号的下降沿被移出。这可确保在 SCLx 为高电平时 SDAx 信号是有效的 (如图 15-9)。

来自主接收器的 \overline{ACK} 脉冲将在 SCLx 输入第 9 个脉冲的上升沿锁存。如果 SDAx 线为高电平 (无 \overline{ACK})，那么表示数据传输已完成。在这种情况下，如果从器件锁存了 \overline{ACK} ，将复位从动逻辑 (复位 SSPxSTAT 寄存器)，同时从器件监视下一个起始位的出现。如果 SDAx 线为低电平 (\overline{ACK})，则必须将接下去要发送的数据装入 SSPxBUF 寄存器。必须通过将 CKP 置 1，再次使能 RC3 或 RD0。

每个数据传输字节都会产生一个 MSSP 中断。SSPxIF 位必须由软件清零，SSPxSTAT 寄存器用于确定字节的状态。SSPxIF 位在第 9 个时钟脉冲的下降沿被置 1。

图 15-8: I²C™ 从动模式接收时序 (SEN = 0, 7 位地址)



PIC18F45J10 系列

图 15-9: I²C™ 从动模式发送时序 (7 位地址)

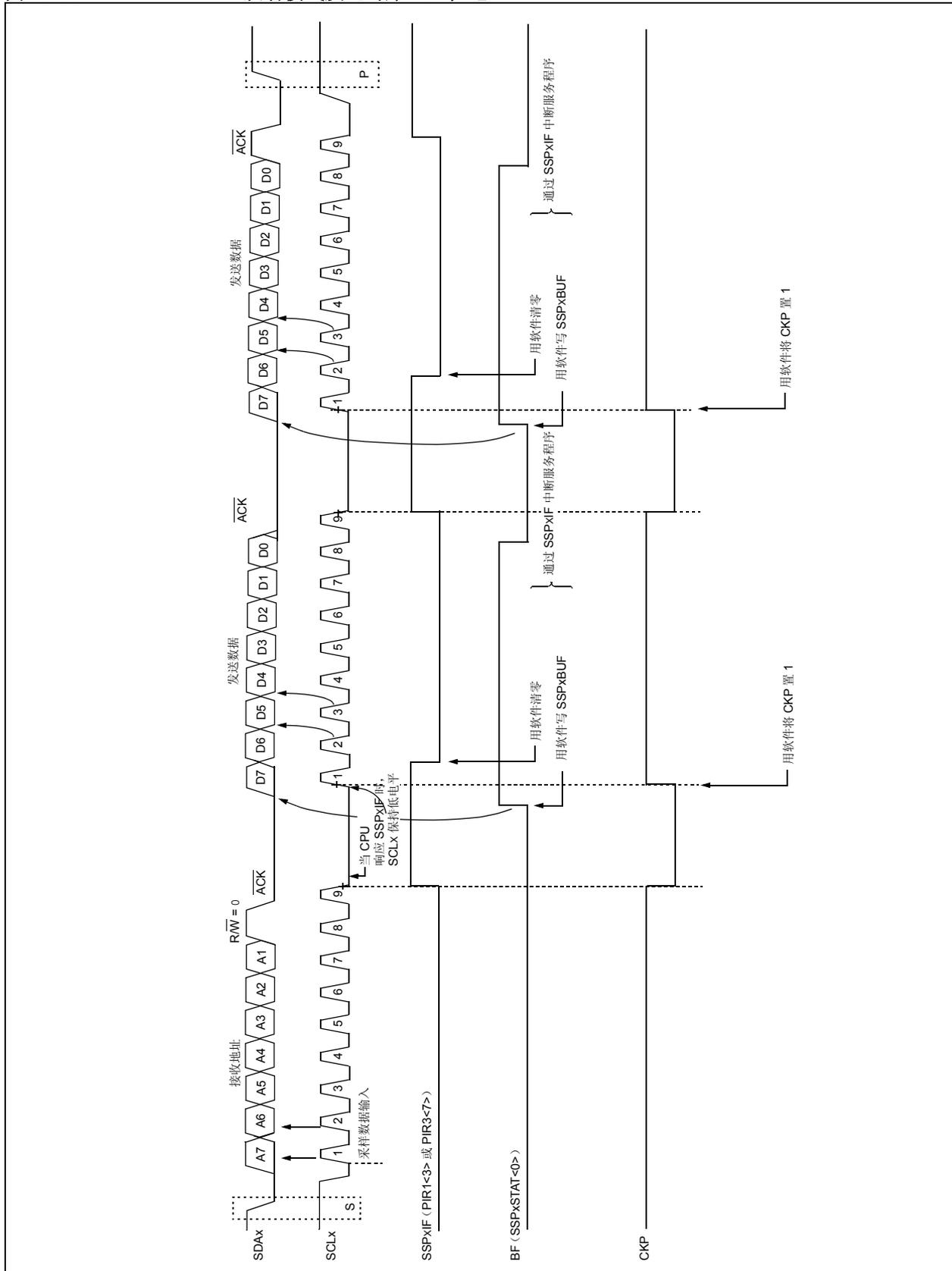
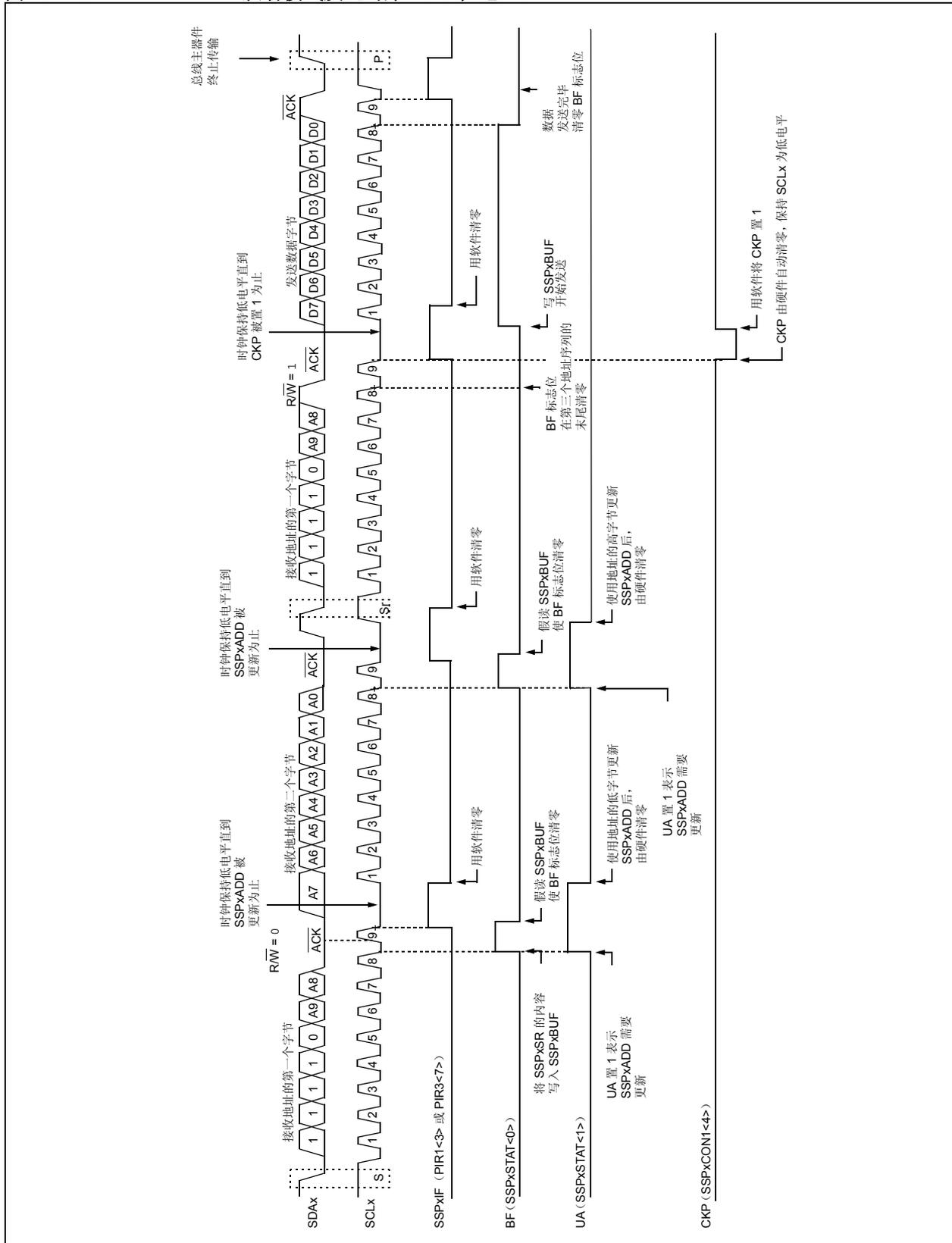


图 15-11: I²C™ 从动模式发送时序 (10 位地址)



15.4.4 时钟延长

7位和10位从动模式都在发送序列上实现了时钟的自动延长。

SEN 位 (SSPxCON2<0>) 允许在接收过程中使能时钟延长。将 SEN 置 1 会导致在每个数据接收序列末尾将 SCLx 引脚保持在低电平。

15.4.4.1 7位从动接收模式 (SEN = 1) 时钟延长

在 7 位从动接收模式下, 在 $\overline{\text{ACK}}$ 序列末尾的第 9 个时钟的下降沿, 如果 BF 置 1, SSPxCON1 寄存器中 CKP 位会自动清零, 强制 SCLx 输出保持低电平。清零 CKP 将迫使 SCLx 线保持低电平。在允许继续接收之前, 必须在用户的 ISR 中将 CKP 位置 1。保持 SCLx 线为低电平可以让用户在主器件发起另一个接收序列之前, 有时间响应 ISR 并读取 SSPxBUF 的内容。这将防止发生缓冲器溢出 (见图 15-13)。

- 注 1:** 如果用户在第 9 个时钟的下降沿前读取了 SSPxBUF 的内容, 使得 BF 被清零, CKP 位就不会被清零, 也不会产生时钟延长。
- 2:** 不管 BF 位的状态如何, CKP 位都可以用软件置 1。用户在下一个接收序列开始之前, 清零 ISR 中的 BF 位时应该小心, 以避免溢出。

15.4.4.2 10位从动接收模式 (SEN = 1) 时钟延长

在 10 位从动接收模式下, 在地址序列中会自动发生时钟延长, 但是 CKP 位不会被清零。在这期间, 如果 UA 位在第 9 个时钟之后置位, 时钟延长就启动了。在接收到 10 位地址的高字节后 UA 位置 1, 然后接收 10 位地址的第二个字节并将 R/W 位清零。在更新 SSPxADD 的过程中释放时钟线。如 7 位模式中描述的那样, 在每个数据接收序列中会发生时钟延长。

- 注:** 如果用户在第 9 个时钟的下降沿出现之前查询 UA 位, 并通过更新 SSPxADD 寄存器清零 UA 位, 而且用户没有提前读取 SSPxBUF 寄存器使 BF 位清零, 则 CKP 位的电平仍然不会被拉低。基于 BF 位状态的时钟延长仅在数据序列中出现, 不会出现在地址序列中。

15.4.4.3 7位从动发送模式时钟延长

如果 BF 位被清零, 7 位从动发送模式将通过在第 9 个时钟的下降沿出现后清零 CKP 位, 来实现时钟延长。不管 SEN 位的状态如何, 这种情况都会发生。

用户的 ISR 必须先将 CKP 位置 1 才可以继续发送。通过保持 SCLx 线为低电平, 用户在主器件发起另一个发送序列之前, 将有时间响应 ISR 并装入 SSPxBUF 的内容 (见图 15-9)。

- 注 1:** 如果用户在第 9 个时钟的下降沿之前就装入 SSPxBUF 的内容, 从而使 BF 位置 1, CKP 位就不会被清零, 也不会发生时钟延长。
- 2:** 不管 BF 位的状态如何, CKP 位都可以用软件置 1。

15.4.4.4 10位从动发送模式时钟延长

正如在 10 位从动接收模式中一样, 在 10 位从动发送模式下, 在前两个地址序列中由 UA 位的状态来控制时钟延长。头两个地址后跟着第三个地址序列, 该地址序列包含 10 位地址的高位和被置为 1 的 R/W 位。在执行完第三个地址序列后, UA 位不置 1, 此时器件配置为发送模式, 正如在 7 位从动发送模式中一样, BF 标志位控制时钟延长 (见图 15-11)。

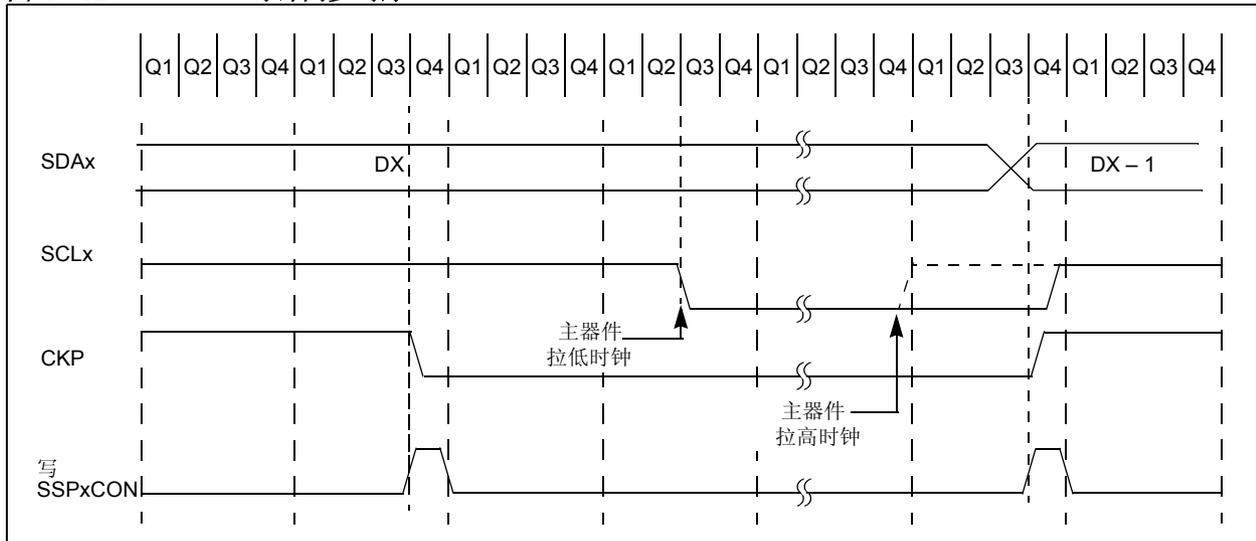
PIC18F45J10 系列

15.4.4.5 时钟同步和 CKP 位

当 CKP 位清零时，SCLx 输出被强制为 0。然而，清零 CKP 位不会将 SCLx 输出拉为低电平，除非已经采样到 SCLx 输出为低电平。因此，CKP 位将不会把 SCLx 线拉为低电平，需由外部 I²C 主器件将 SCLx 线拉低。SCLx

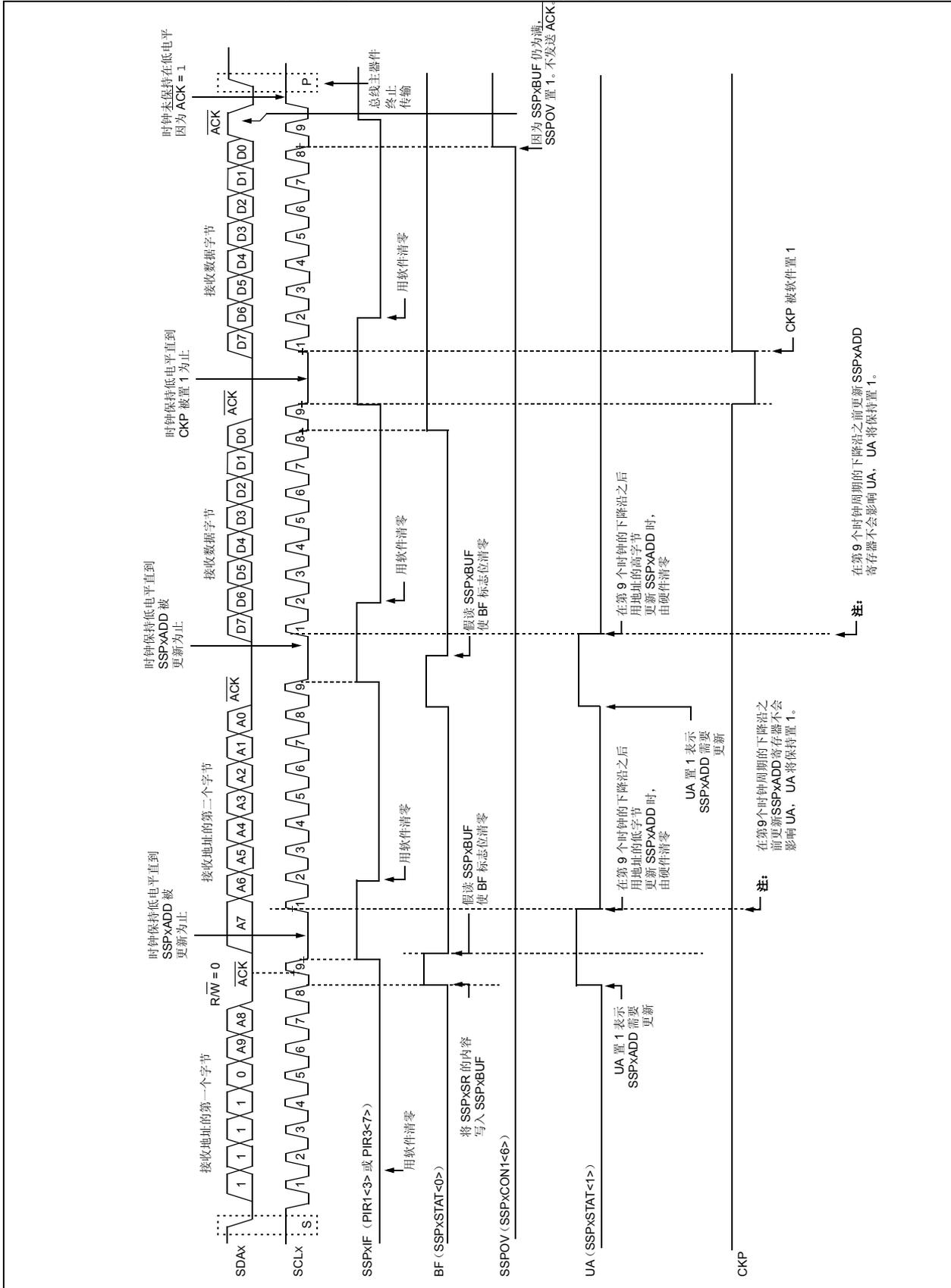
输出将保持低电平，直到 CKP 位置 1，且 I²C 总线上的其他器件将 SCLx 电平拉高为止。这可以确保对 CKP 位的写操作不会违反 SCLx 的最小高电平时间要求（见图 15-12）。

图 15-12: 时钟同步时序



PIC18F45J10 系列

图 15-14: I²C™ 从动模式接收时序 (SEN = 1, 10 位地址)



15.4.5 广播呼叫地址支持

在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答响应。

广播呼叫地址是由 I²C 协议为特定目的保留的八个地址之一。它由全 0 组成，且 R/W = 0。

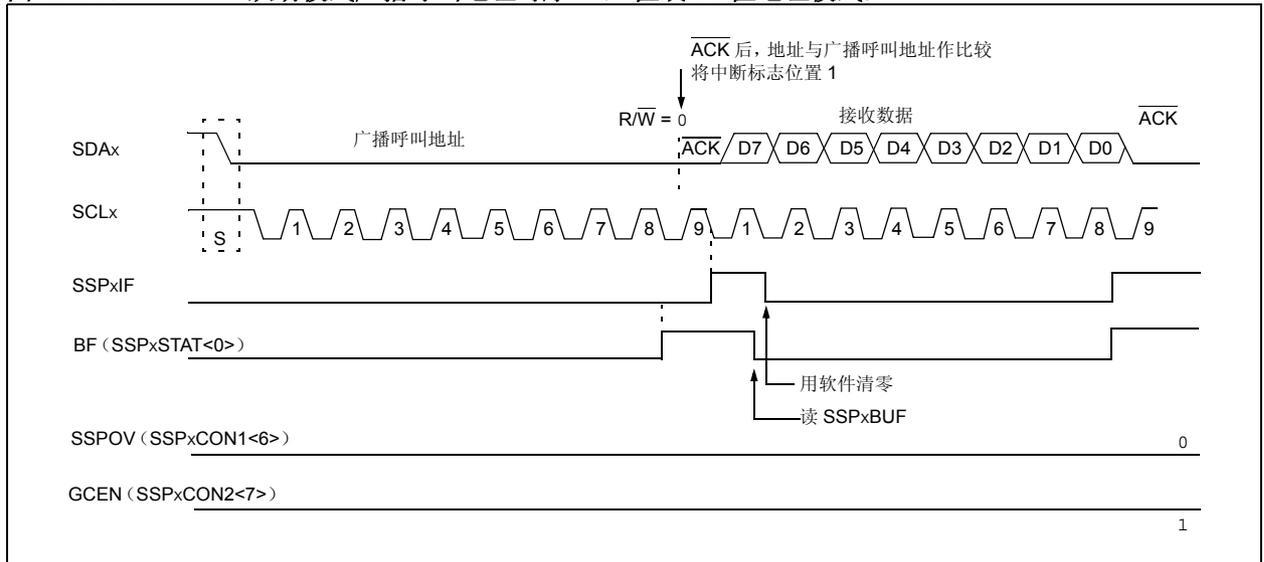
广播呼叫使能位 (GCEN) 使能 (SSPxCON2<7> 置 1) 时，即可识别广播呼叫地址。检测到起始位后，8 位数据会移入 SSPxSR，同时将该地址与 SSPxADD 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配，SSPxSR 的值将传输到 SSPxBUF，BF 标志位 (第 8 位) 置 1，并且 SSPxIF 中断标志位在第 9 位 (ACK 位) 的下降沿置 1。

当响应中断时，可以通过读取 SSPxBUF 的内容来判断中断源。该值可以用于判断地址是特定器件的还是一个广播呼叫地址。

在 10 位模式下，需要更新 SSPxADD 来进行地址的后半部分匹配，同时 UA 位 (SSPxSTAT<1>) 置 1。如果 GCEN 位置 1 时采样到广播呼叫地址，同时从器件被配置为 10 位地址模式，则不再需要地址的后半部分，也不会将 UA 位置 1，从器件将在应答后开始接收数据 (图 15-15)。

图 15-15: 从动模式广播呼叫地址时序 (7 位或 10 位地址模式)



PIC18F45J10 系列

15.4.6 主控模式

通过将 SSPxCON1 中的相应 SSPM 位置位和清零，同时将 SSPEN 位置 1，可以使能主控模式。在主控模式下，SCLx 和 SDAx 线由 MSSP 硬件控制。

主控模式通过在检测到启动和停止条件时产生中断来工作。停止 (P) 位和起始 (S) 位在复位时或禁止 MSSP 模块时清零。当 P 位置 1 时，可以取得 I²C 总线的控制权；否则总线处于空闲状态，且 P 位和 S 位都为零。

在固件控制的主控模式下，用户代码根据起始和停止位条件执行所有的 I²C 总线操作。

一旦使能主控模式，用户即可选择以下 6 项操作：

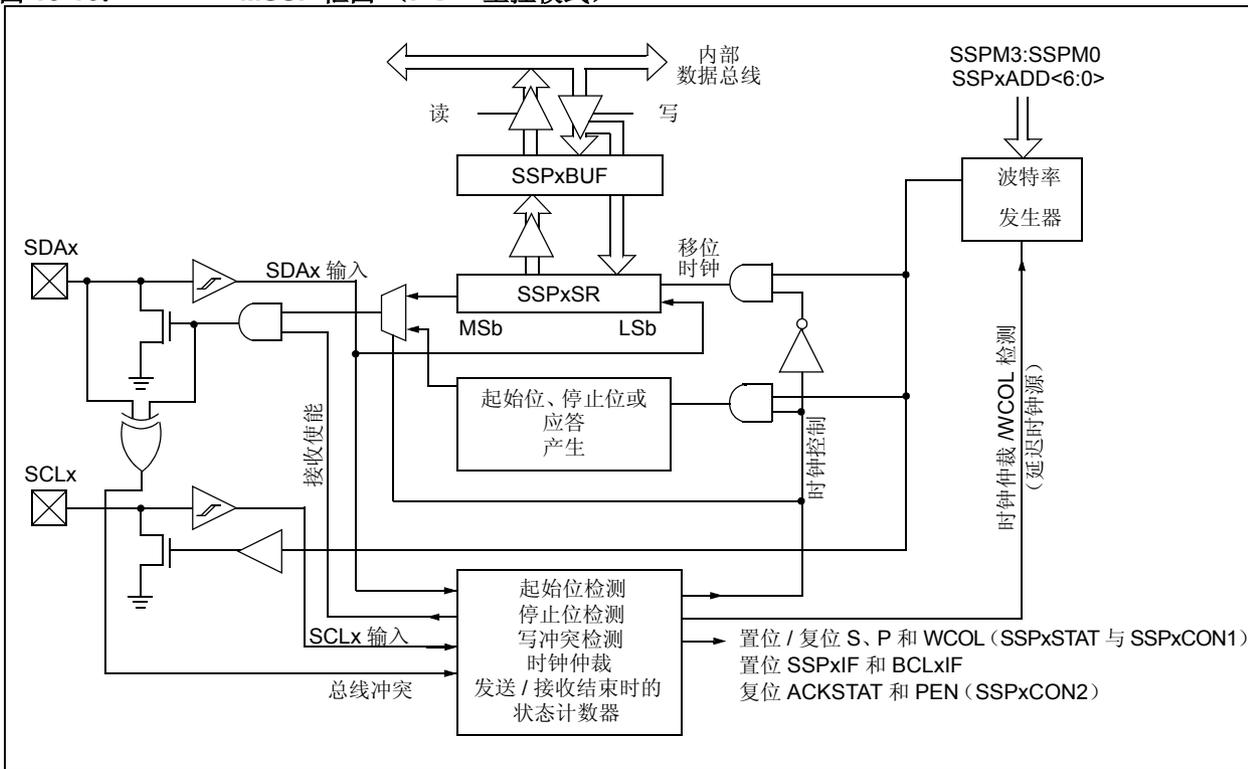
1. 在 SDAx 和 SCLx 上发出一个启动条件。
2. 在 SDAx 和 SCLx 上发出一个重复启动条件。
3. 写入 SSPxBUF 寄存器，开始数据 / 地址的发送。
4. 配置 I²C 端口接收数据。
5. 在接收到数据字节后产生应答条件。
6. 在 SDAx 和 SCLx 上产生停止条件。

注： 当配置为 I²C 主控模式时，MSSP 模块不允许事件排队。例如，在启动条件结束前，不允许用户发出另一个启动条件并立即写 SSPxBUF 寄存器以发起传输。这种情况下，将不会写入 SSPxBUF。WCOL 位将被置 1，这表明没有发生对 SSPxBUF 的写操作。

下列事件会使 MSSP 中断标志位 SSPxIF 置 1（如果允许 MSSP 中断，则产生中断）：

- 启动条件
- 停止条件
- 数据传输字节已发送 / 已接收
- 应答发送
- 重复启动

图 15-16: MSSP 框图 (I²C™ 主控模式)



15.4.6.1 I²C 主控模式工作方式

主器件产生所有串行时钟脉冲和启动 / 停止条件。以停止条件或重复启动条件结束传输。因为重复启动条件也是下一次串行传输的开始，因此不会释放 I²C 总线。

在主控发送器模式下，串行数据通过 SDA_x 输出，而串行时钟由 SCL_x 输出。发送的第一个字节包括接收器件的地址（7 位）和读 / 写（R/W）位。在这种情况下，R/W 位将是逻辑 0。一次发送 8 位串行数据。每发送一个字节，会收到一个应答位。输出启动和停止条件，表明串行传输的开始和结束。

在主控接收模式下，发送的第一个字节包括发送器件的地址（7 位）和 R/W 位。在这种情况下，R/W 位将是逻辑 1。因此，发送的第一个字节是一个 7 位从器件地址，后面跟 1 表示接收。串行数据通过 SDA_x 接收，而串行时钟由 SCL_x 输出。每次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件分别表明发送的开始和结束。

在 I²C 模式下，在 SPI 模式中使用的波特率发生器被用于将 SCL_x 时钟频率设置为 100 kHz、400 kHz 或 1 MHz。更多详情，请参见第 15.4.7 节“波特率”。

下面是一个典型的发送事件序列：

1. 用户通过将启动使能位 SEN (SSPxCON2<0>) 置 1 产生启动条件。
2. SSPxIF 位置 1。在进行任何其他操作前，MSSP 模块将等待所需的启动时间。
3. 用户将从器件地址装入 SSPxBUF 进行发送。
4. 地址从 SDA_x 引脚移出，直到发送完所有 8 位为止。
5. MSSP 模块移入来自从器件的 ACK 位，并将它的值写入 SSPxCON2 寄存器 (SSPxCON2<6>)。
6. MSSP 模块在第 9 个时钟周期的末尾将 SSPxIF 位置 1，产生一个中断。
7. 用户将 8 位数据装入 SSPxBUF。
8. 数据从 SDA_x 引脚移出，直到发送完所有 8 位为止。
9. MSSP 模块移入来自从器件的 ACK 位，并将它的值写入 SSPxCON2 寄存器 (SSPxCON2<6>)。
10. MSSP 模块在第 9 个时钟的末尾将 SSPxIF 位置 1，产生一个中断。
11. 用户通过将停止使能位 PEN (SSPxCON2<2>) 置 1 产生停止条件。
12. 一旦停止条件完成，将产生一个中断。

PIC18F45J10 系列

15.4.7 波特率

在 I²C 主控模式下，波特率发生器（BRG）的重载值位于 SSPxADD 寄存器的低 7 位（图 15-17）。当发生对 SSPxBUF 的写操作时，波特率发生器将自动开始计数。BRG 会递减计数至 0，然后停止直到下次重载为止。BRG 会在每个指令周期（Tcy）中的 Q2 和 Q4 时钟周期上进行两次减计数。在 I²C 主控模式下，会自动重载 BRG。

如果指定操作完成（即，在传输的最后一个数据位后面跟着 ACK），内部时钟将自动停止计数，SCLx 引脚将保持在其最后的状态。

表 15-3 显示了不同的指令周期下的时钟频率以及装入 SSPxADD 的 BRG 值。

15.4.7.1 波特率和模块的相互关系

因为 MSSP1 和 MSSP2 是独立的模块，所以它们可以以不同的波特率同时在 I²C 主控模式下工作。这是通过每个模块使用不同的 BRG 重载值实现的。

由于此模式的基本时钟源来自系统时钟，对系统时钟的任何更改将会同等程度地影响这两个模块。通过更改 BRG 重载值也许可以将一个或两个波特率改回到前一个值。

图 15-17: 波特率发生器框图

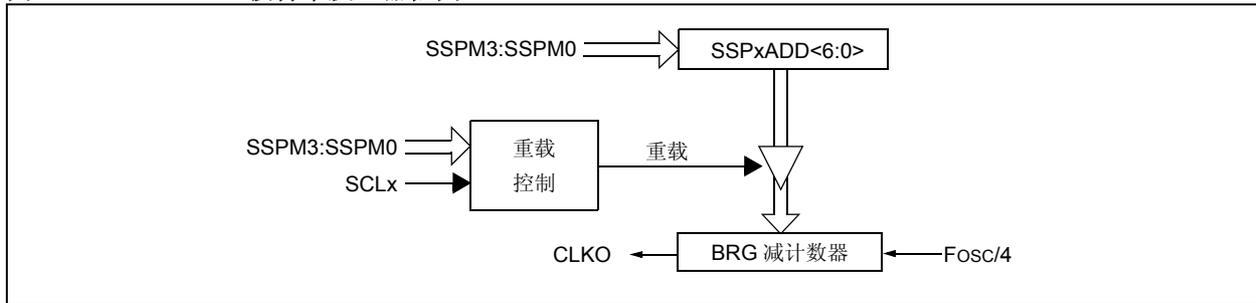


表 15-3: I²C™ 时钟速率和 BRG

Fcy	Fcy * 2	BRG 值	Fscl (两次 BRG 计满返回)
10 MHz	20 MHz	18h	400 kHz ⁽¹⁾
10 MHz	20 MHz	1Fh	312.5 kHz
10 MHz	20 MHz	63h	100 kHz
4 MHz	8 MHz	09h	400 kHz ⁽¹⁾
4 MHz	8 MHz	0Ch	308 kHz
4 MHz	8 MHz	27h	100 kHz
1 MHz	2 MHz	02h	333 kHz ⁽¹⁾
1 MHz	2 MHz	09h	100 kHz
1 MHz	2 MHz	00h	1 MHz ⁽¹⁾

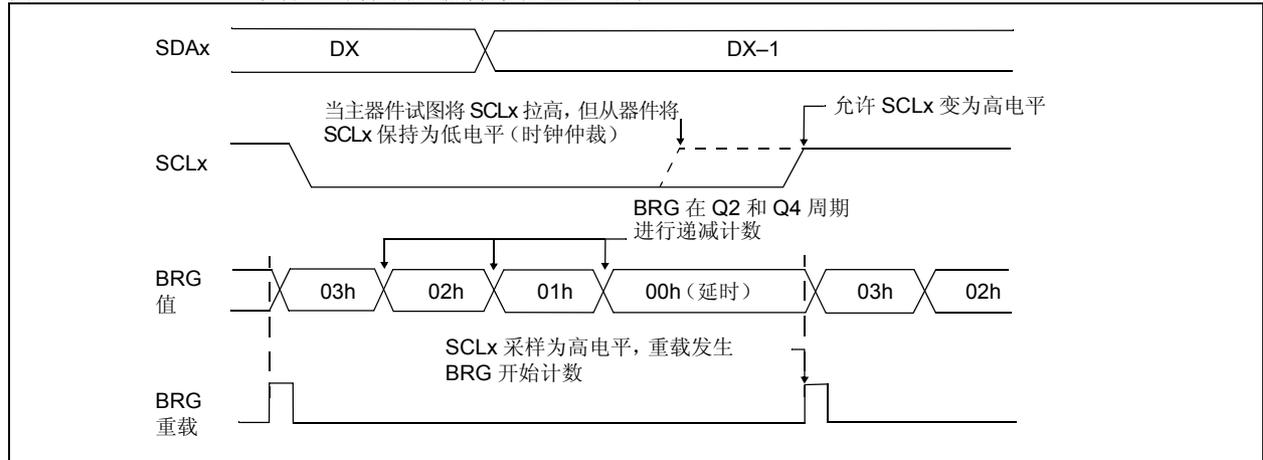
注 1: 虽然 I²C™ 接口各方面都不符合 400 kHz I²C 规范（该规范适用于大于 100 kHz 的频率），但在需要较高频率的应用场合可以小心使用。

15.4.7.2 时钟仲裁

如果在任何接收、发送或重复启动 / 停止条件期间，主器件拉高了 SCLx 引脚（允许 SCLx 引脚悬空为高电平），就会发生时钟仲裁。如果允许 SCLx 引脚悬空为高电平，波特率发生器（BRG）将暂停计数直到实际采样到 SCLx 引脚为高电平为止。当 SCLx 引脚采样为高

电平时，波特率发生器将被重新装入 SSPxADD<6:0> 的内容并开始计数。这可以保证当外部器件将时钟拉低时，SCLx 始终保持至少一个 BRG 计满返回周期的高电平（图 15-18）。

图 15-18: 带有时钟仲裁的波特率发生器时序



PIC18F45J10 系列

15.4.8 I²C 主控模式启动条件时序

要发起启动条件，用户应将启动使能位 SEN (SSPxCON2<0>) 置 1。当 SDAx 和 SCLx 引脚都采样为高电平时，波特率发生器重新装入 SSPxADD<6:0> 的内容并开始计数。当波特率发生器发生超时 (TBRG) 时，如果 SCLx 和 SDAx 都采样为高电平，则 SDAx 引脚被驱动为低电平。当 SCLx 为高电平时，将 SDAx 驱动为低电平就是启动条件，将使 S 位 (SSPxSTAT<3>) 置 1。随后波特率发生器重新装入 SSPxADD<6:0> 的内容并恢复计数。当波特率发生器超时 (TBRG) 时，SEN 位 (SSPxCON2<0>) 将自动被硬件清零。波特率发生器暂停工作，SDAx 线保持低电平，启动条件结束。

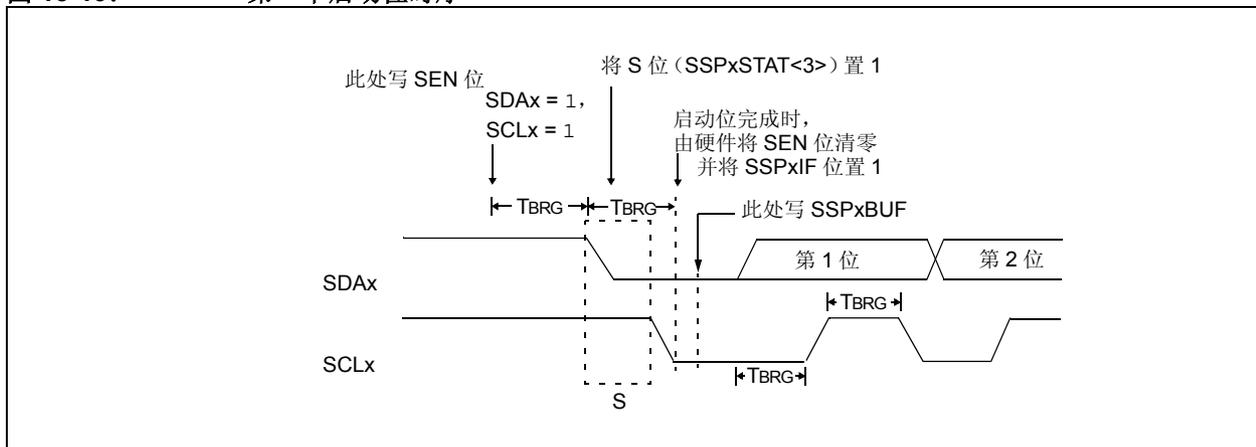
注： 如果在启动条件开始时，SDAx 和 SCLx 引脚已经采样为低电平，或者在启动条件期间，SCLx 在 SDAx 线被驱动为低电平之前已经采样为低电平，则会发生总线冲突。总线冲突中断标志位 BCLxIF 置 1，启动条件中止，I²C 模块复位到空闲状态。

15.4.8.1 WCOL 状态标志

当启动序列进行时，如果用户写 SSPxBUF，则 WCOL 被置 1，同时缓冲器内容不变（写操作无效）。

注： 由于不允许事件排队，在启动条件结束之前，不能对 SSPxCON2 的低 5 位进行写操作。

图 15-19: 第一个启动位时序



15.4.9 I²C 主控模式重复启动条件时序

将 RSEN 位 (SSPxCON2<1>) 编程为高电平, 并且 I²C 逻辑模块处于空闲状态时, 就会产生重复启动条件。当 RSEN 位置 1 时, SCLx 引脚被拉为低电平。当 SCLx 引脚采样为低电平时, 波特率发生器装入 SSPxADD<6:0> 的内容, 并开始计数。在一个波特率发生器计数周期 (TBRG) 内 SDAx 引脚被释放 (其引脚电平被拉高)。当波特率发生器超时, 如果 SDAx 采样为高电平, SCLx 引脚将被拉高。当 SCLx 引脚采样为高电平时, 波特率发生器将被重新装入 SSPxADD<6:0> 的内容并开始计数。SDAx 和 SCLx 必须在一个计数周期 TBRG 内采样为高电平。随后将 SDAx 引脚拉为低电平 (SDAx = 0) 并保持一个计数周期 TBRG, 同时 SCLx 为高电平。然后 RSEN 位 (SSPxCON2<1>) 将自动清零, 波特率发生器不会重载, SDAx 引脚保持低电平。一旦在 SDAx 和 SCLx 引脚上检测到启动条件, S 位 (SSPxSTAT<3>) 将被置 1。直到波特率发生器超时后, SSPxIF 位才会置 1。

- 注 1:** 有任何其他事件进行时, 编程设置对 RSEN 无效。
- 注 2:** 在重复启动条件期间, 下列事件将会导致总线冲突:
- 当 SCLx 由低电平变为高电平时, SDAx 采样为低电平。
 - 在 SDAx 被拉低之前, SCLx 变为低电平。这表示可能有另一个主器件正尝试发送数据 1。

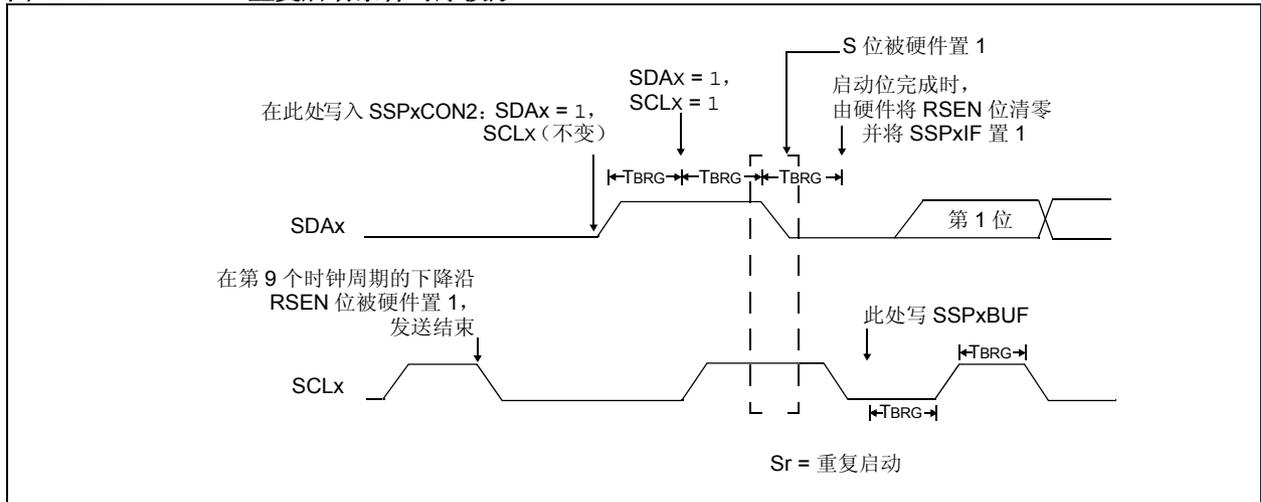
一旦 SSPxIF 位被置 1, 用户便可以在 7 位地址模式下将 7 位地址写入 SSPxBUF, 或者在 10 位地址模式下写入默认的 8 位地址字节。当发送完第一个 8 位并接收到一个 ACK 后, 用户可以发送另外 8 位地址 (10 位地址模式下) 或 8 位数据 (7 位地址模式下)。

15.4.9.1 WCOL 状态标志

当重复启动序列进行时, 如果用户写 SSPxBUF, 则 WCOL 被置 1, 同时缓冲器内容不变 (写操作无效)。

- 注:** 由于不允许事件排队, 在重复启动条件结束之前, 不能对 SSPxCON2 的低 5 位进行写操作。

图 15-20: 重复启动条件时序波形



15.4.10 I²C 主控模式发送

发送一个数据字节、一个 7 位地址或一个 10 位地址的另一半，都可以通过写一个值到 SSPxBUF 寄存器来实现。该操作将使缓冲器满标志位 BF 置 1，并且波特率发生器开始计数，同时启动下一次发送。在 SCLx 的下降沿有效后（见数据保持时间规范参数 106），地址 / 数据的每一位将被移出至 SDAx 引脚。在一个波特率发生器计满返回计数周期（TBRG）内，SCLx 保持低电平。数据应该在 SCLx 释放为高电平前保持有效（见数据建立时间规范参数 107）。当 SCLx 引脚被释放为高电平时，它将在整个 TBRG 中保持高电平状态。在此期间以及下一个 SCLx 下降沿之后的一段时间内，SDAx 引脚上的数据必须保持稳定。在第 8 位被移出（第 8 个时钟周期的下降沿）之后，BF 标志位清零，同时主器件释放 SDAx。此时如果发生地址匹配或是数据被正确接收，被寻址的从器件将在第 9 位时以一个 ACK 位响应。ACK 的状态在第 9 个时钟周期的下降沿写入 ACKDT 位。主器件接收到应答之后，应答状态位 ACKSTAT 会被清零；如果未收到应答，则该位被置 1。第 9 个时钟之后，SSPxIF 位会置 1，主控时钟（波特率发生器）暂停，直到下一个数据字节装入 SSPxBUF 为止，SCLx 引脚保持低电平，SDAx 保持不变（图 15-21）。

在写 SSPxBUF 之后，地址的每一位在 SCLx 的下降沿被移出，直至地址的所有 7 位和 R/W 位都被移出为止。在第 8 个时钟的下降沿，主器件将 SDAx 引脚拉为高电平，以允许从器件发出应答响应。在第 9 个时钟的下降沿，主器件通过采样 SDAx 引脚来判断地址是否被从器件识别。ACK 位的状态被装入 ACKSTAT 状态位（SSPxCON2<6>）。在发送地址的第 9 个时钟下降沿之后，SSPxIF 置 1，BF 标志位清零，波特率发生器关闭直到下一次写 SSPxBUF，且 SCLx 引脚保持低电平，允许 SDAx 引脚悬空。

15.4.10.1 BF 状态标志

在发送模式下，BF 位（SSPxSTAT<0>）在 CPU 写 SSPxBUF 时置 1，在所有 8 位数据移出后清零。

15.4.10.2 WCOL 状态标志

如果用户在发送过程中（即，SSPxSR 仍在移出数据字节时）写 SSPxBUF，则 WCOL 置 1，且在写 SSPBUF 之后的两个 Tcy 内缓冲器内容不变（写操作无效）。如果在两个 Tcy 内重新写 SSPBUF，则 WCOL 位置 1，并更新 SSPBUF。这可能导致传输被破坏。

在每次写 SSPBUF 之后，用户应该确认 WCOL 位时清零状态以确保传输是正确的。在所有情况下，WCOL 必须由软件清零。

15.4.10.3 ACKSTAT 状态标志

在发送模式下，当从器件发送应答响应（ACK = 0）时，ACKSTAT 位（SSPxCON2<6>）清零；当从器件没有应答（ACK = 1）时，该位置 1。从器件在识别出其地址（包括广播呼叫地址）或正确接收数据后，会发送一个应答。

15.4.11 I²C 主控模式接收

通过编程接收使能位 RCEN（SSPxCON2<3>）使能主控模式接收。

注： RCEN 位置 1 前，MSSP 模块必须处于空闲状态，否则对 RCEN 位置 1 将无效。

波特率发生器开始计数，每次计满返回时，SCLx 引脚的状态发生改变（由高变低或由低变高），数据被移入 SSPxSR。第 8 个时钟的下降沿之后，接收使能标志位自动清零，SSPxSR 的内容装入 SSPxBUF，BF 标志位置 1，SSPxIF 标志位置 1，波特率发生器暂停计数，SCLx 保持为低电平。此时 MSSP 处于空闲状态，等待下一条命令。当 CPU 读缓冲器时，BF 标志位将自动清零。通过将应答序列使能位 ACKEN（SSPxCON2<4>）置 1，用户可以在接收结束后发送应答位。

15.4.11.1 BF 状态标志

接收时，当将地址或数据字节从 SSPxSR 装入 SSPxBUF 时，BF 位置 1；在读 SSPxBUF 寄存器时清零。

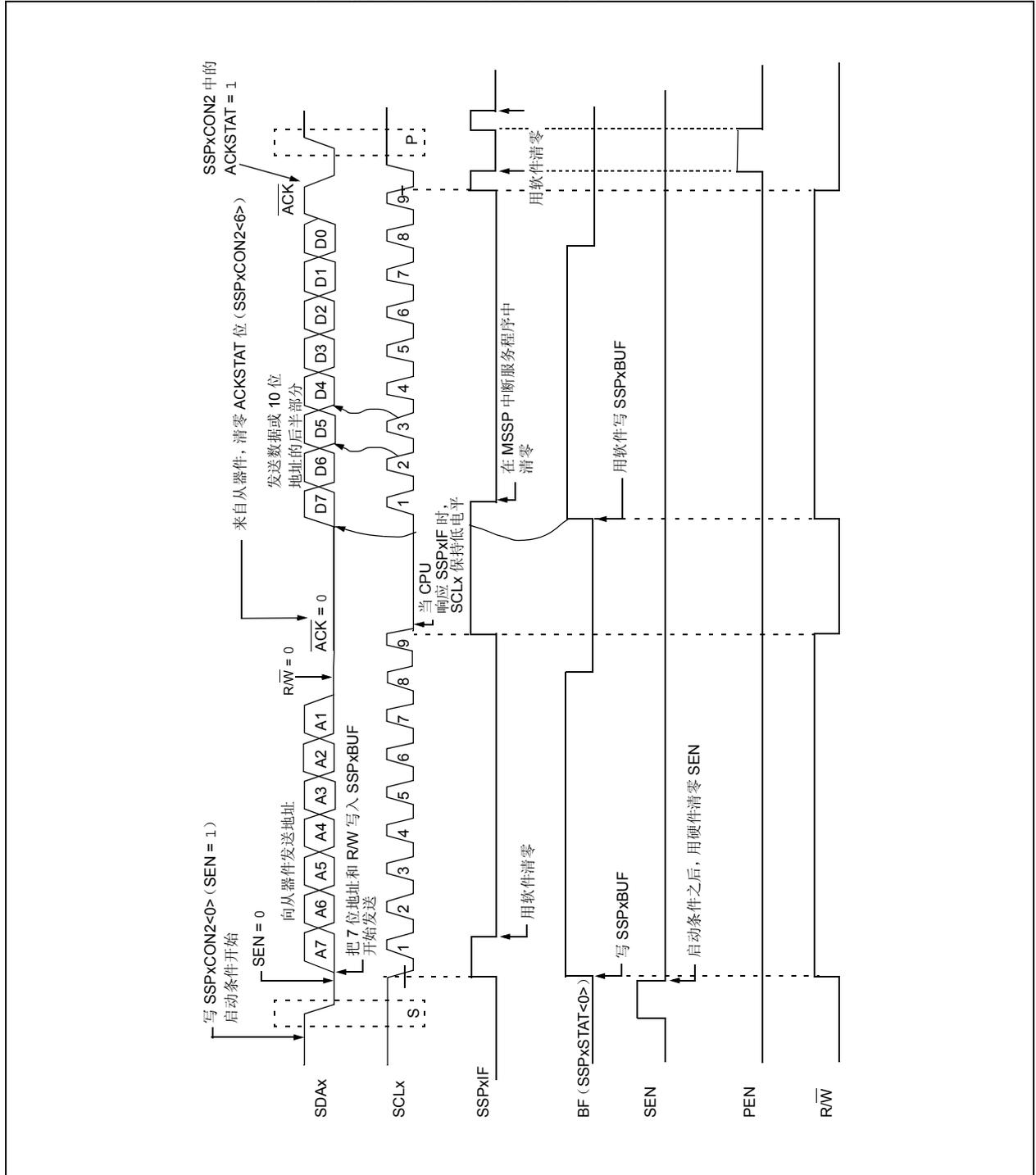
15.4.11.2 SSPOV 状态标志

接收时，当 SSPxSR 接收到 8 位数据时，SSPOV 位置 1，BF 标志位已经在上一次接收时置 1。

15.4.11.3 WCOL 状态标志

如果用户在接收过程中（即，SSPxSR 仍在移入数据字节时）写 SSPxBUF，则 WCOL 位置 1，缓冲器内容不变（写操作无效）。

图 15-21: I²C™ 主控模式发送时序 (7 位或 10 位地址)



15.4.12 应答序列时序

将应答序列使能位 **ACKEN** (**SSPxCON2<4>**) 置 1 即可使能应答序列。当该位被置 1 时, **SCLx** 引脚被拉低, 应答数据位的内容出现在 **SDAx** 引脚上。如果用户希望产生一个应答, 则应该将 **ACKDT** 位清零; 否则, 用户应该在应答序列开始前将 **ACKDT** 位置 1。然后波特率发生器进行一个计满返回周期 (**TBRG**) 的计数, 随后 **SCLx** 引脚电平被拉高。当 **SCLx** 引脚采样为高电平时 (时钟仲裁), 波特率发生器再进行一个 **TBRG** 周期的计数。然后 **SCLx** 引脚被拉低。在这之后, **ACKEN** 位自动清零, 波特率发生器关闭, **MSSP** 模块进入空闲模式 (图 15-23)。

15.4.12.1 WCOL 状态标志

如果用户在应答序列进行过程中试图写 **SSPxBUF**, 则 **WCOL** 将置 1, 缓冲器的内容不会改变 (写操作无效)。

15.4.13 停止条件时序

通过将停止序列使能位 **PEN** (**SSPxCON2<2>**) 置 1, 在接收 / 发送结束后, **SDAx** 引脚上将产生停止位。在接收 / 发送结束后, **SCLx** 线在第 9 个时钟的下降沿之后保持低电平。当 **PEN** 位置 1 时, 主器件将 **SDAx** 线置为低电平。当 **SDAx** 线采样为低电平时, 波特率发生器被重载并递减计数至 0。当波特率发生器超时, **SCLx** 引脚被拉为高电平, 在一个 **TBRG** (波特率发生器计满返回周期) 之后, **SDAx** 引脚将被拉高。当 **SDAx** 引脚采样为高电平且 **SCLx** 也是高电平时, **P** 位 (**SSPxSTAT<4>**) 置 1。一个 **TBRG** 之后, **PEN** 位清零, 同时 **SSPxIF** 位置 1 (图 15-24)。

15.4.13.1 WCOL 状态标志

如果用户在停止序列过程中试图写 **SSPxBUF**, 则 **WCOL** 位将置 1, 缓冲器的内容不会改变 (写操作无效)。

图 15-23: 应答序列时序波形

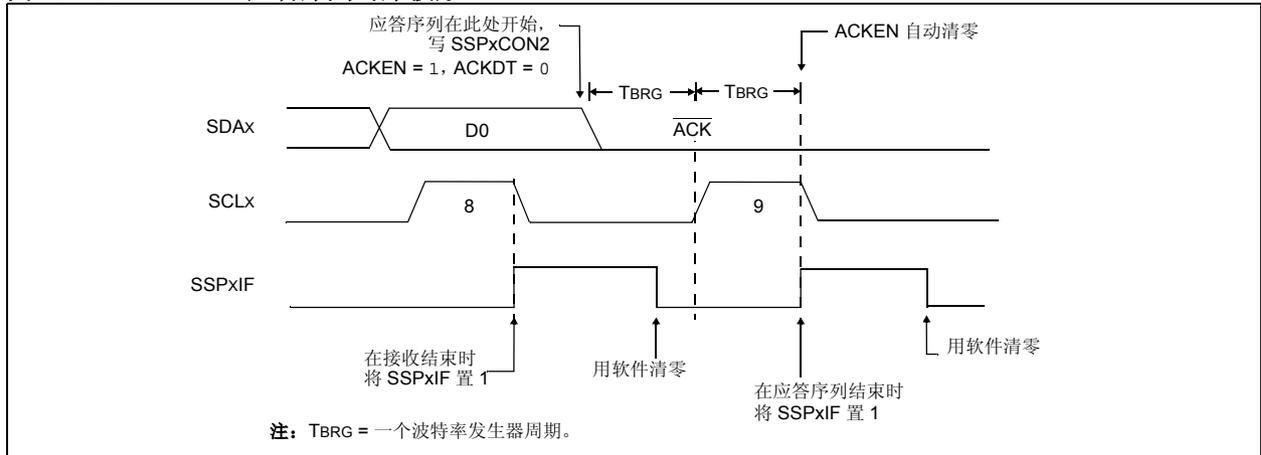
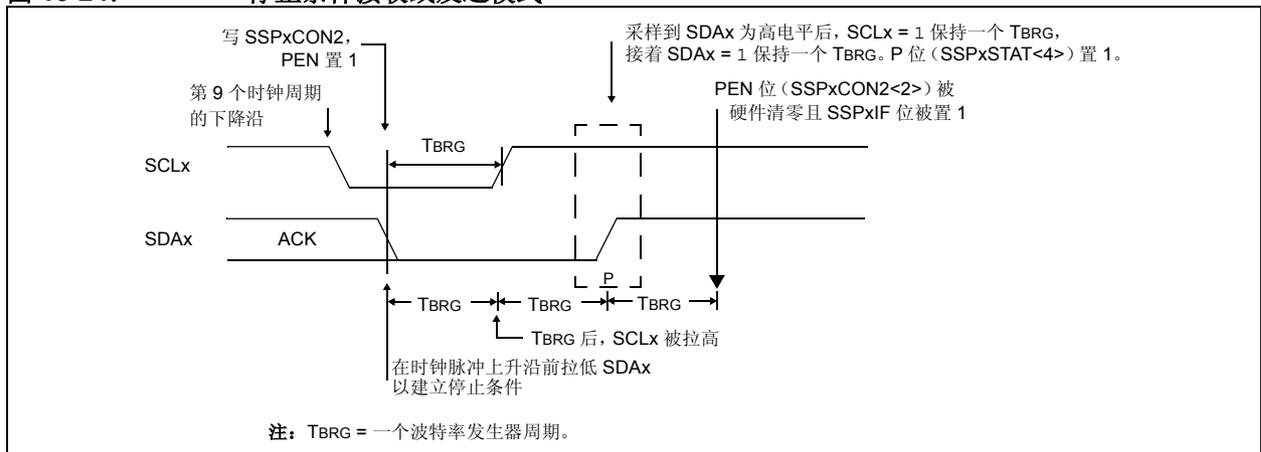


图 15-24: 停止条件接收或发送模式



PIC18F45J10 系列

15.4.14 休眠模式下的工作方式

在休眠模式下，I²C 模块能够接收地址或数据。并且在地址匹配或字节传输完成后，如果允许 MSSP 中断，将唤醒处理器。

15.4.15 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

15.4.16 多主机模式

在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。停止 (P) 位和启动 (S) 位在复位时或禁止 MSSP 模块时清零。当 P 位 (SSPxSTAT<4>) 置位时，可以取得 I²C 总线的控制权；否则总线处于空闲状态，且 P 位和 S 位清零。当总线忙时，如果出现停止条件，则将产生中断 (若允许 MSSP 中断)。

在多主机模式下工作时，必须监视 SDAx 线来进行仲裁，查看信号电平是否为期望的输出电平。此检查由硬件完成，其结果放在 BCLxIF 位。

仲裁可能失败的状态是：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

15.4.17 多主机通信、总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。当主器件将地址 / 数据位输出到 SDAx 引脚时，如果一个主器件通过将 SDAx 引脚悬空为高电平以在 SDAx 上输出 1，而另一个主器件输出 0，就会发生总线仲裁。当 SCLx 悬空为高电平时，数据应该保持稳定。如果 SDAx 引脚上期望的数据是 1，而实际在 SDAx 引脚上采样到的数据是 0，则发生了总线冲突。主器件将把总线冲突中断标志位 BCLx1F 置 1，并将 I²C 端口复位到空闲状态 (图 15-25)。

如果在发送过程中发生总线冲突，则发送停止，BF 标志位清零，SDAx 和 SCLx 线被拉高，并且允许对 SSPxBUF 进行写操作。当执行完总线冲突中断服务程序后，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

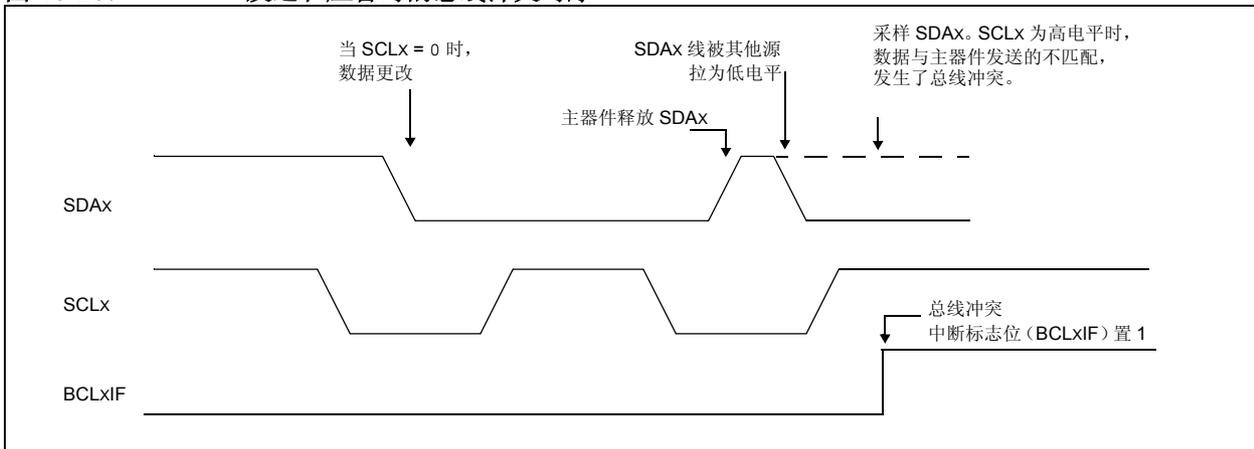
如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突，则该条件被中止，SDAx 和 SCLx 线被拉高，SSPxCON2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

主器件将继续监视 SDAx 和 SCLx 引脚。如果出现停止条件，SSPxIF 位将被置 1。

无论发生总线冲突时发送的进度如何，写 SSPxBUF 都会从第一个数据位开始发送数据。

在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。SSPxSTAT 寄存器中的 P 位置 1 时，可以获得 I²C 总线的控制权，否则总线空闲且 S 和 P 位清零。

图 15-25: 发送和应答时的总线冲突时序



15.4.17.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- 在启动条件开始时，SDAx 或 SCLx 被采样为低电平（图 15-26）。
- SDAx 被拉低之前，SCLx 采样为低电平（图 15-27）。

在启动条件期间，SDAx 和 SCLx 引脚都会被监视。

如果 SDAx 引脚已经是低电平，或 SCLx 引脚已经是低电平，则：

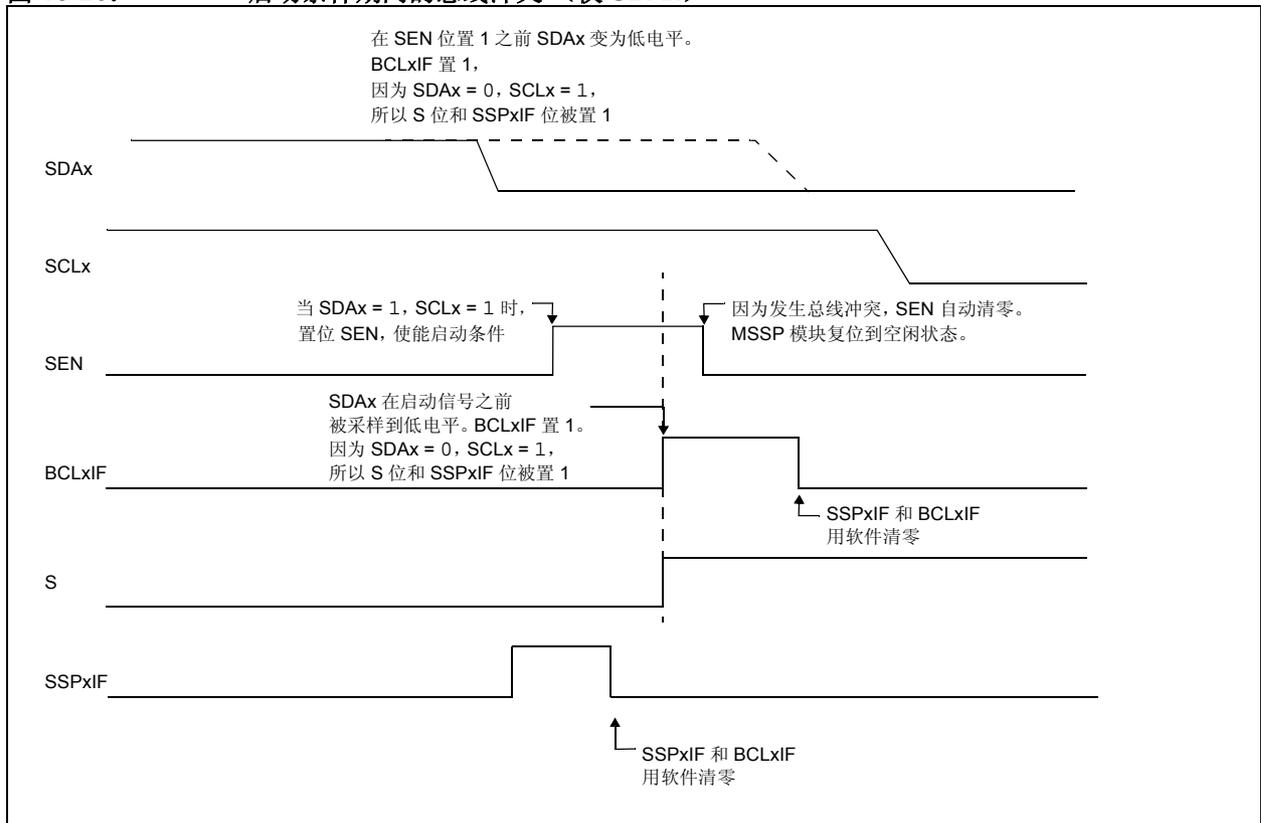
- 中止启动条件，
- BCLxIF 标志位置 1，且
- MSSP 模块复位为空闲状态（图 15-26）。

启动条件从 SDAx 和 SCLx 引脚被拉高开始。当 SDAx 引脚采样为高电平时，波特率发生器装入 SSPxADD<6:0> 的值并递减计数到 0。如果在 SDAx 为高电平时，SCLx 引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图发送一个数据 1。

如果 SDAx 引脚在该计数周期内采样为低电平，则 BRG 复位，同时 SDAx 线保持原值（图 15-28）。但是，如果 SDAx 引脚采样为 1，SDAx 引脚将在 BRG 计数结束时被置为低电平。接着，波特率发生器被重新装入值并递减计数至 0。在此期间，如果 SCLx 引脚采样到 0，则没有发生总线冲突。在 BRG 计数结束时，SCLx 引脚被拉为低电平。

注： 在启动条件期间不会发生总线冲突是因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此总是有一个主器件先于另一个主器件将 SDAx 拉低。但是这一情况不会引起总线冲突，因为允许两个主器件对启动条件后的第一个地址进行仲裁。如果地址是相同的，将继续对数据部分、重复启动条件或停止条件进行仲裁。

图 15-26: 启动条件期间的总线冲突（仅 SDAx）



PIC18F45J10 系列

图 15-27: 启动条件期间的总线冲突 (SCLx = 0)

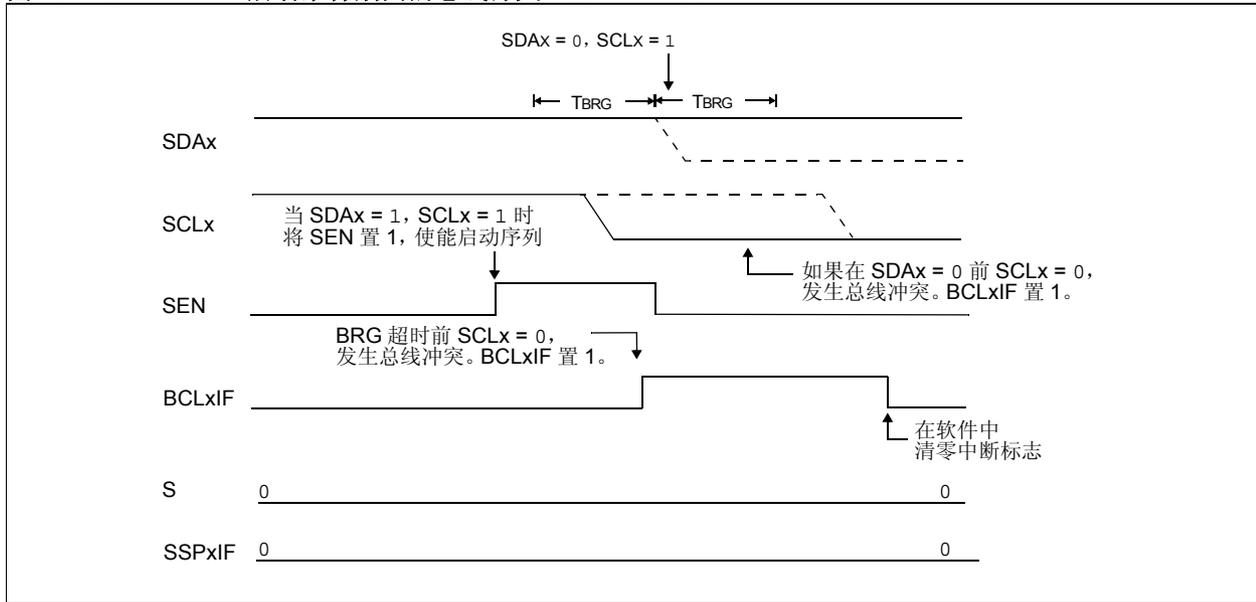
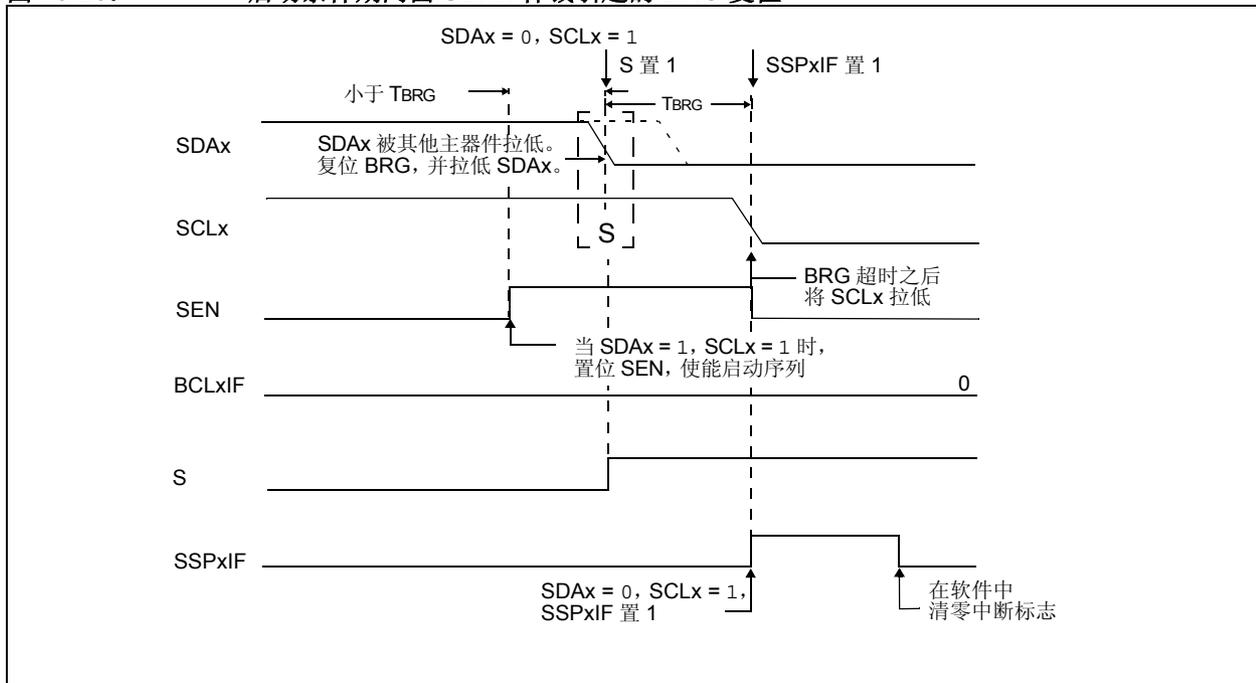


图 15-28: 启动条件期间由 SDAx 仲裁引起的 BRG 复位



15.4.17.2 重复启动条件期间的总线冲突

在下列情况中，重复启动条件期间会发生总线冲突：

- 在 SCLx 由低电平变为高电平的过程中，SDAx 采样到低电平。
- 在 SDAx 被拉为低电平之前，SCLx 变为低电平，表示另一个主器件正试图发送一个数据 1。

当用户拉高 SDAx 并允许该引脚悬空时，BRG 装入 SSPxADD<6:0> 中的值并递减计数至 0。接着 SCLx 引脚被置为高电平，当 SCLx 采样到高电平时，对 SDAx 引脚进行采样。

如果 SDAx 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据 0，见图 15-29）。如果 SDAx 采样为高电平，则 BRG 被重新装入值并开始计数。如果 SDAx 在 BRG 超时之前从高电平变为低电平，则没有发生总线冲突，因为两个主器件不可能精确地在同一时刻将 SDAx 拉低。

如果 SCLx 在 BRG 超时之前从高电平变为低电平，且 SDAx 尚未变为低电平，表示发生了总线冲突。在此情况下，在重复启动条件期间另一个主器件正试图发送一个数据 1（见图 15-30）。

如果在 BRG 超时结束时 SCLx 和 SDAx 都仍然是高电平，则 SDAx 引脚被拉低，BRG 重新装入值并开始计数。在计数结束时，不管 SCLx 引脚的状态如何，SCLx 引脚都被拉低，重复启动条件结束。

图 15-29: 重复启动条件期间的总线冲突（情形 1）

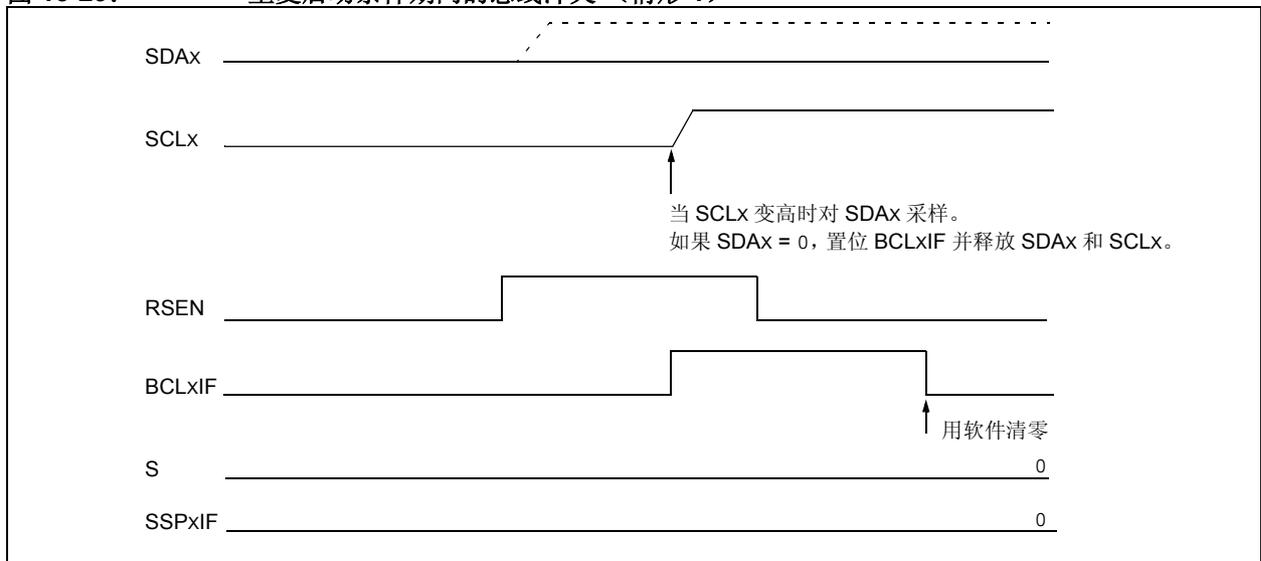
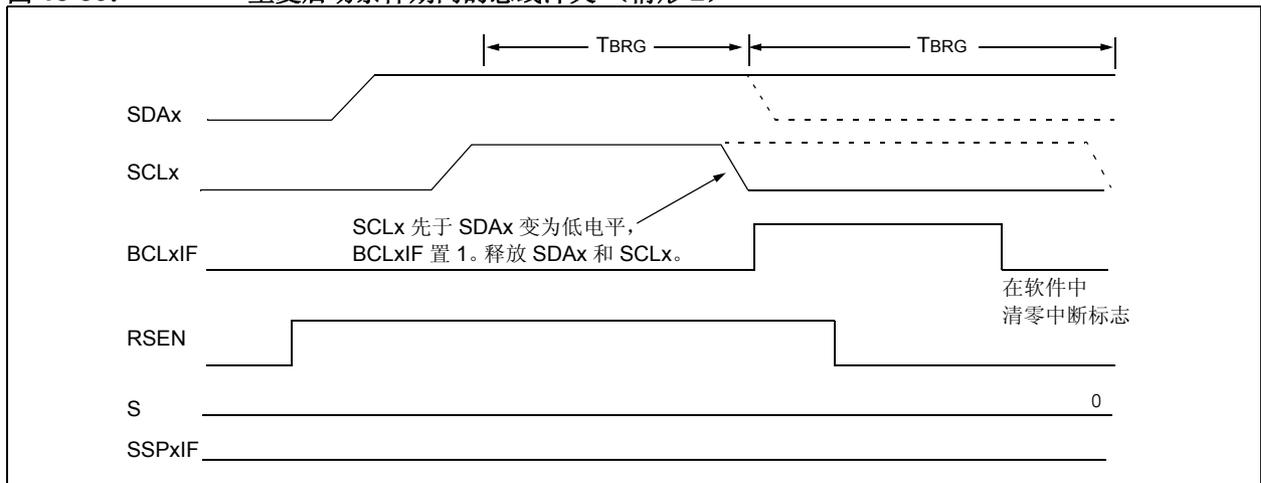


图 15-30: 重复启动条件期间的总线冲突（情形 2）



PIC18F45J10 系列

15.4.17.3 停止条件期间的总线冲突

以下事件会导致停止条件期间的总线冲突：

- a) SDAx 已被拉高并允许悬空为高电平之后，SDAx 在 BRG 超时后被采样到低电平。
- b) SCLx 引脚被拉高之后，SCLx 在 SDAx 变成高电平之前被采样到低电平。

停止条件从 SDAx 被拉低开始。当 SDAx 采样为低电平时，SCLx 引脚就可以悬空为高电平。当引脚被采样到高电平时（时钟仲裁），波特率发生器装入 SSPxADD<6:0> 的值并递减计数到 0。BRG 超时后，采样 SDAx。如果 SDAx 采样到低电平，则已发生总线冲突。这是因为另一个主器件正试图发送一个数据 0（图 15-31）。如果 SCLx 引脚在允许 SDAx 悬空为高电平前被采样到低电平，也会发生总线冲突。这是另一个主器件正试图发送一个数据 0 的另一种情况（图 15-32）。

图 15-31: 停止条件期间的总线冲突（情形 1）

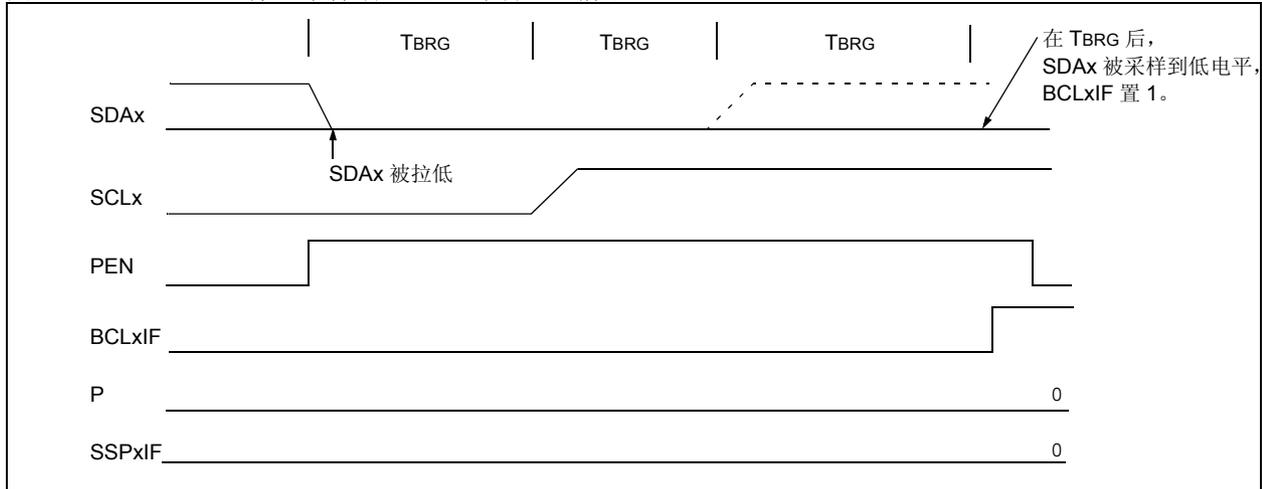


图 15-32: 停止条件期间的总线冲突（情形 2）

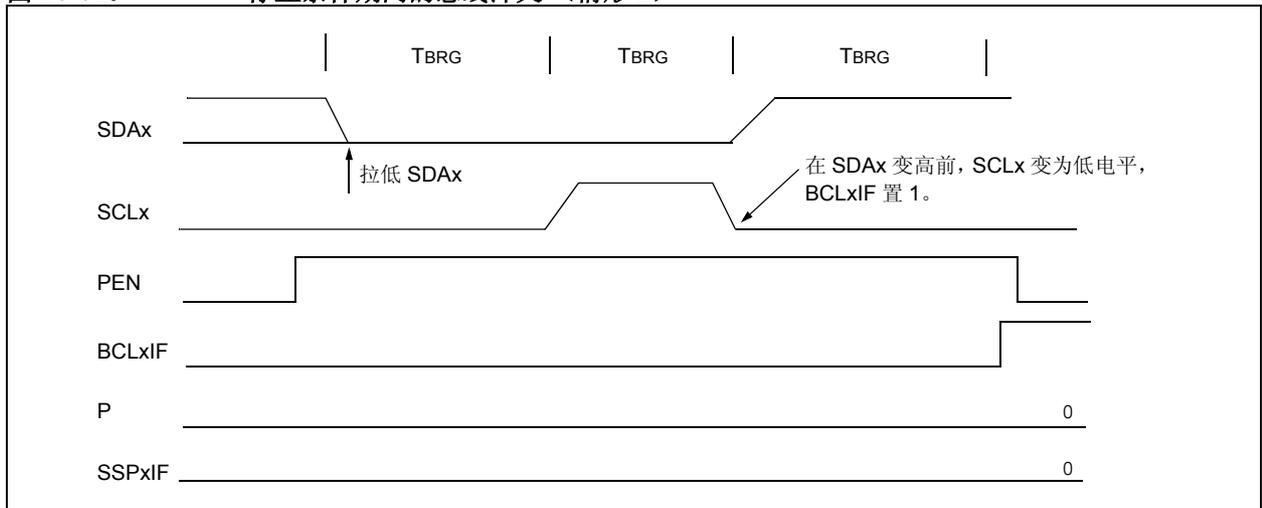


表 15-4: 与 I²C™ 工作模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
PIR3	SSP2IF	BCL2IF	—	—	—	—	—	—	45
PIE3	SSP2IE	BCL2IE	—	—	—	—	—	—	45
IPR3	SSP2IP	BCL2IP	—	—	—	—	—	—	45
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	46
TRISD ⁽¹⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	46
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								44
SSP1ADD	MSSP1 地址寄存器 (I ² C™ 从动模式) MSSP1 波特率重载寄存器 (I ² C™ 主控模式)								44
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	44
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	44
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	44
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								46
SSP2ADD	MSSP2 地址寄存器 (I ² C™ 从动模式) MSSP2 波特率重载寄存器 (I ² C™ 主控模式)								46
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	46
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	46
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	46

图注: — = 未用 (读为 0)。在 I²C™ 模式下, MSSP 模块不使用阴影单元。

注 1: 在 28 引脚器件中这些寄存器和 / 或位不被使用, 并且应该读为 0。

PIC18F45J10 系列

注:

16.0 增强型通用同步 / 异步收发器 (EUSART)

增强型通用同步 / 异步收发器 (Enhanced Universal Synchronous Asynchronous Receiver Transmitter, EUSART) 模块是两个串行 I/O 模块之一。(通常, 也将 USART 称为串行通信接口或 SCI。) EUSART 可以被配置为能与 CRT 终端和个人计算机等外设通信的全双工异步系统; 也可以被配置成能够与 A/D 或 D/A 集成电路、串行 EEPROM 等外设通信的半双工同步系统。

增强型 USART 模块实现了更多的功能, 包括自动波特率检测和校准以及在接收到“同步间隔”字符和发送 12 位间隔字符时自动唤醒。这些功能使 EUSART 模块成为局域互连网络 (Local Interconnect Network, LIN) 总线系统非常理想的选择。

可将 EUSART 配置为以下几种工作模式:

- 带有以下功能的全双工异步模式:
 - 字符接收自动唤醒
 - 自动波特率校准
 - 12 位间隔字符发送
- 半双工同步主控模式 (时钟极性可选)
- 半双工同步从动模式 (时钟极性可选)

增强型 USART 的引脚与 PORTC 复用。为了把 RC6/TX/CK 和 RC7/RX/DT 引脚配置为 EUSART:

- SPEN (RCSTA<7>) 位必须置 1 (= 1)
- TRISC<7> 位必须置 1 (= 1)
- TRISC<6> 位必须置 1 (= 1)

注: EUSART 在需要时会自动将引脚从输入重新配置为输出。

增强型 USART 模块的操作是由 3 个寄存器控制的:

- 发送状态和控制寄存器 (TXSTA)
- 接收状态和控制寄存器 (RCSTA)
- 波特率控制寄存器 (BAUDCON)

这些寄存器将在后面几页中的寄存器 16-1、寄存器 16-2 和寄存器 16-3 中详细介绍。

PIC18F45J10 系列

寄存器 16-1:

TXSTA: 发送状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

bit 7 **CSRC:** 时钟源选择位

异步模式:

忽略。

同步模式:

1 = 主控模式 (时钟来自内部 BRG)

0 = 从动模式 (时钟来自外部时钟源)

bit 6 **TX9:** 9 位发送使能位

1 = 选择 9 位发送

0 = 选择 8 位发送

bit 5 **TXEN:** 发送使能位

1 = 使能发送

0 = 禁止发送

注: 同步模式下 SREN/CREN 的优先级高于 TXEN。

bit 4 **SYNC:** EUSART 模式选择位

1 = 同步模式

0 = 异步模式

bit 3 **SENDB:** 发送间隔字符位

异步模式:

1 = 在下一次发送时发送“同步间隔”字符 (在完成时由硬件清零)

0 = “同步间隔”字符发送完成

同步模式:

忽略。

bit 2 **BRGH:** 高波特率选择位

异步模式:

1 = 高速

0 = 低速

同步模式:

在此模式下未使用。

bit 1 **TRMT:** 发送移位寄存器状态位

1 = TSR 空

0 = TSR 满

bit 0 **TX9D:** 发送数据的第 9 位

该位可以是地址 / 数据位或奇偶校验位。

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 16-2: RCSTA: 接收状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7						bit 0	

- bit 7 **SPEN:** 串行端口使能位
 1 = 使能串行端口 (将 RX/DT 和 TX/CK 引脚配置为串行端口引脚)
 0 = 禁止串行端口 (保持在复位状态)
- bit 6 **RX9:** 9 位接收使能位
 1 = 选择 9 位接收
 0 = 选择 8 位接收
- bit 5 **SREN:** 单字节接收使能位
异步模式:
 忽略。
同步主控模式:
 1 = 使能单字节接收
 0 = 禁止单字节接收
 此位在接收完成后清零。
同步从动模式:
 忽略。
- bit 4 **CREN:** 连续接收使能位
异步模式:
 1 = 使能接收器
 0 = 禁止接收器
同步模式:
 1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 比 SREN 的优先级高)
 0 = 禁止连续接收
- bit 3 **ADDEN:** 地址检测使能位
9 位异步模式 (RX9 = 1):
 1 = 当 RSR<8> 置 1 时, 使能地址检测、允许中断并装载接收缓冲器
 0 = 禁止地址检测、接收所有字节并且第 9 位可用作奇偶校验位
9 位异步模式 (RX9 = 0):
 忽略。
- bit 2 **FERR:** 帧错误位
 1 = 帧错误 (可以通过读 RCREG 寄存器刷新并接收下一个有效字节)
 0 = 无帧错误
- bit 1 **OERR:** 溢出错误位
 1 = 溢出错误 (可以通过清零 CREN 位清零)
 0 = 无溢出错误
- bit 0 **RX9D:** 接收数据的第 9 位
 该位可以是地址 / 数据位或奇偶校验位, 并且必须由用户固件计算得到。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

寄存器 16-3: BAUDCON: 波特率控制寄存器

R/W-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

- bit 7 **ABDOVF**: 自动波特率采样溢出状态位
 1 = 在自动波特率检测模式下出现了 BRG 溢出
 (必须用软件清零)
 0 = 没有发生 BRG 溢出
- bit 6 **RCIDL**: 接收操作空闲状态位
 1 = 接收操作处于空闲状态
 0 = 接收操作处于活动状态
- bit 5 **未用**: 读为 0
- bit 4 **SCKP**: 同步时钟极性选择位
异步模式:
 在此模式下未使用。
同步模式:
 1 = 空闲状态时钟 (CK) 为高电平
 0 = 空闲状态时钟 (CK) 为低电平
- bit 3 **BRG16**: 16 位波特率寄存器使能位
 1 = 16 位波特率发生器——SPBRGH 和 SPBRG
 0 = 8 位波特率发生器——仅 SPBRG (兼容模式), 忽略 SPBRGH 值
- bit 2 **未用**: 读为 0
- bit 1 **WUE**: 唤醒使能位
异步模式:
 1 = EUSART 将继续采样 RX 引脚——中断在下降沿产生, 在下一个上升沿由硬件清零该位
 0 = 未监测 RX 引脚或检测到了上升沿
同步模式:
 在此模式下未使用。
- bit 0 **ABDEN**: 自动波特率检测使能位
异步模式:
 1 = 在下一个字符使能波特率检测。需要收到“同步”字段 (55h), 完成时由硬件清零。
 0 = 禁止波特率检测或检测已完成
同步模式:
 在此模式下未使用。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

16.1 波特率发生器 (BRG)

BRG 是一个专用的 8 位或 16 位发生器，支持 EUSART 的异步和同步模式。默认情况下，BRG 工作在 8 位模式下，将 BRG16 位 (BAUDCON<3>) 置 1 可选择 16 位模式。

SPBRGH:SPBRG 寄存器对控制一个独立运行定时器的周期。在异步模式下，BRGH (TXSTA<2>) 位和 BRG16 (BAUDCON<3>) 位也控制波特率。在同步模式下，BRGH 位会被忽略。表 16-1 所示为不同 EUSART 模式的波特率计算公式，但仅适用于主控模式 (由内部产生时钟信号)。

给定目标波特率和 Fosc 的情况下，可以使用表 16-1 中的公式计算 SPBRGH:SPBRG 寄存器中的近似整数值，从而确定波特率误差。例 16-1 给出了计算示例。表 16-2

中给出了不同异步模式下典型的波特率和误差值。使用高波特率 (BRGH = 1) 或 16 位 BRG 有利于减少波特率误差，或者在快速振荡频率条件下实现低波特率。

向 SPBRGH:SPBRG 寄存器写入新值会引起 BRG 定时器复位 (或清零)。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

16.1.1 在功耗管理模式下的操作

器件时钟用于产生所需的波特率。当进入一种功耗管理模式时，新时钟源可能会工作在一个不同的频率下。这可能需要调整 SPBRG 寄存器对中的值。

16.1.2 采样

择多检测电路对 RX 引脚采样三次，以判定 RX 引脚上出现的是高电平还是低电平。

表 16-1: 波特率公式

配置位			BRG/EUSART 模式	波特率计算公式
SYNC	BRG16	BRGH		
0	0	0	8 位 / 异步	$F_{osc}/[64(n+1)]$
0	0	1	8 位 / 异步	$F_{osc}/[16(n+1)]$
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	$F_{osc}/[4(n+1)]$
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注: x = 任意值, n = SPBRGH:SPBRG 寄存器对的值

PIC18F45J10 系列

例 16-1: 计算波特率误差

针对工作在异步模式下，工作频率 Fosc 为 16 MHz，采用 8 位 BRG，目标波特率为 9600bps 的器件：

$$\text{目标波特率} = \text{Fosc}/(64 ([\text{SPBRGH}:\text{SPBRG}] + 1))$$

求解 SPBRGH:SPBRG:

$$X = ((\text{Fosc}/\text{目标波特率})/64) - 1$$

$$= ((16000000/9600)/64) - 1$$

$$= [25.042] = 25$$

$$\text{波特率计算结果} = 16000000/(64 (25 + 1))$$

$$= 9615$$

$$\text{误差} = (\text{波特率计算结果} - \text{目标波特率}) / \text{目标波特率}$$

$$= (9615 - 9600)/9600 = 0.16\%$$

表 16-2: 与波特率发生器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注： — = 未用（读为 0）。BRG 未使用阴影单元。

PIC18F45J10 系列

表 16-3: 异步模式下的波特率

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-	1	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

PIC18F45J10 系列

表 16-3: 异步模式下的波特率 (续)

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20,000 MHz			Fosc = 10,000 MHz			Fosc = 8.000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	0.300	0.04	832	0.300	-0.16	415	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20,000 MHz			Fosc = 10,000 MHz			Fosc = 8.000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.647	-2.12	16

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)	实际波特率 (Kbps)	误差 %	SPBRG 值 (10 进制)
0.3	0.300	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

16.1.3 自动波特率检测

增强型 USART 模块支持波特率自动检测和校准。此功能仅在异步模式下当 WUE 位清零时有效。

只要接收到起始位并且 ABDEN 位已置 1，就会开始自动波特率检测过程（图 16-1）。波特率计算采用自平均的方式。

在自动波特率检测（Auto-Baud Rate Detect, ABD）模式下，BRG 的时钟是反向的。不是由 BRG 为 RX 输入信号提供时钟源，而是由 RX 信号为 BRG 定时。在 ABD 模式下，内部波特率发生器被用作计数器来计算输入的串行字节流的位间隔时间。

一旦 ABDEN 位置 1，状态机就会将 BRG 清零并寻找起始位。为了正确计算比特率，自动波特率检测必须接收到一个值为 55h（ASCII 字符 U，也是 LIN 总线的同步字符）的字节。为了尽量减少输入信号不对称所造成的影响，在接收低位和高位的时间内都要进行测量。在起始位后，SPBRG 使用预先选择的时钟源，在 RX 引脚上的第一个上升沿开始向上计数。在 RX 引脚传输了 8 个位，或在检测到第 5 个上升沿后，会将相应 BRG 周期内累计的值保存在 SPBRGH:SPBRG 寄存器对中。当第 5 个时钟沿出现时（应与停止位对应），ABDEN 位会自动清零。

如果发生了 BRG 计满返回（从 FFFFh 到 0000h 的溢出），那么该事件将会在 ABDOMF 状态位（BAUDCON<7>）有所反映。当 BRG 计满返回时，该位由硬件置 1，用户可用软件将其置 1 或清零。在发生计满返回事件后，继续保持 ABD 模式，ABDEN 位保持置 1（图 16-2）。

当校准波特率周期时，BRG 寄存器时钟频率为预配置时钟频率的 1/8。请注意 BRG 时钟将由 BRG16 和 BRGH 位配置。无论 BRG16 位的设置如何，SPBRG 和 SPBRGH 都将被用作 16 位计数器。用户通过检查 SPBRGH 寄存器中的值是否为 00h，可以验证在 8 位模式下是否发生了进位。表 16-4 所示为 BRG 计数器的时钟速率。

当产生 ABD 序列时，EUSART 状态机保持在空闲状态。一旦在 RX 上检测到第 5 个上升沿，中断标志位 RCIF 就会置 1。要清零中断标志位 RCIF，需要读取 RCREG 中的值。应丢弃 RCREG 的值。

- 注 1:** 如果 WUE 位与 ABDEN 位同时置 1，自动波特率检测会在间隔字符之后的字节开始。
- 2:** 由用户来判断进入的字符波特率是否处于所选 BRG 时钟源范围内。由于位误差率的原因，某些振荡频率和 EUSART 波特率的组合是无法实现的。在使用自动波特率检测功能时，必须考虑系统总体时序和通信波特率。

表 16-4: BRG 计数器时钟速率

BRG16	BRGH	BRG 计数器时钟
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

注: 在产生 ABD 序列时，不管 BRG16 的设置如何，SPBRG 和 SPBRGH 都被用作 16 位计数器。

16.1.3.1 ABD 和 EUSART 发送

由于在 ABD 采集期间 BRG 时钟是反向的，所以在 ABD 期间不能使用 EUSART 发送器。这意味着只要 ABDEN 位置 1，就不能写入 TXREG。用户还应确保在发送期间 ABDEN 不能为置 1 状态，否则可能会导致无法预料的 EUSART 操作。

PIC18F45J10 系列

图 16-1: 自动波特率计算

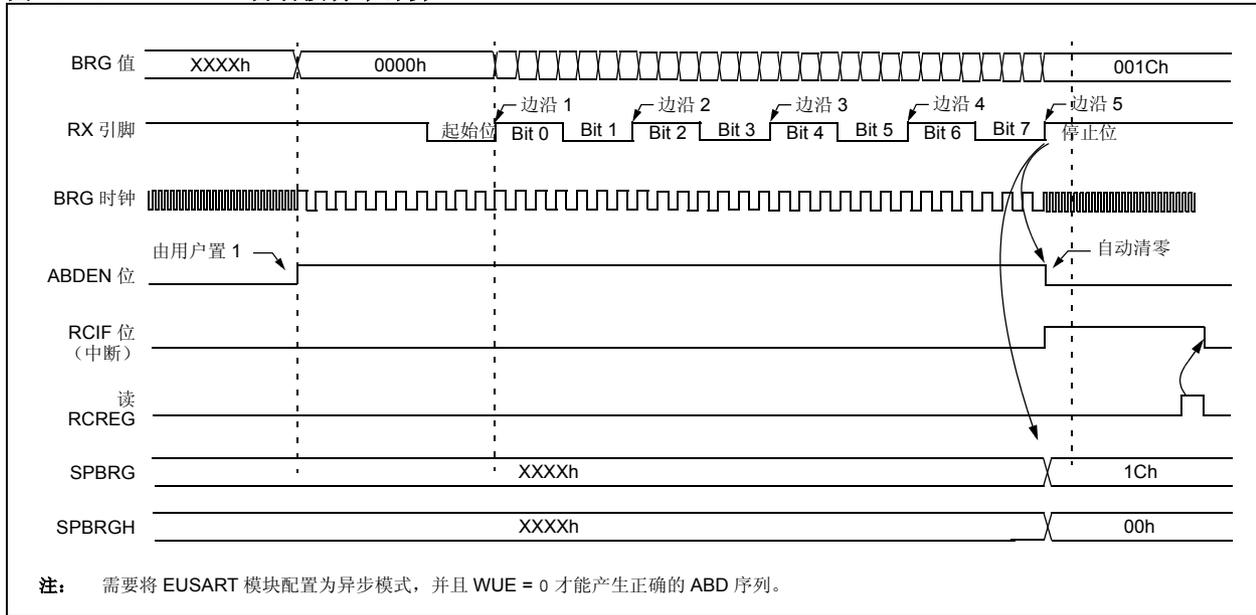
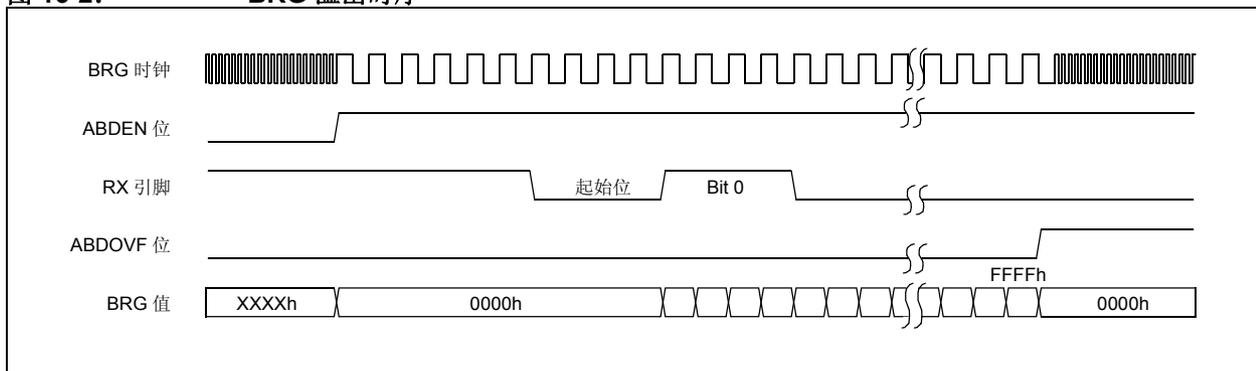


图 16-2: BRG 溢出时序



16.2 EUSART 异步模式

通过将 SYNC (TXSTA<4>) 位清零可选择异步工作模式。在此模式下, EUSART 使用标准的“不归零”(Non-Return-to-Zero, NRZ) 格式 (1 个起始位, 8 个或 9 个数据位, 1 个停止位)。最常用的数据格式为 8 位。片上专用 8 位 /16 位波特率发生器可借助于振荡器产生标准波特率频率。

EUSART 首先发送和接收的是最低有效位。EUSART 的发送器和接收器在功能上是独立的, 但采用相同的数据格式和波特率。波特率发生器可以根据 BRGH 和 BRG16 位 (TXSTA<2> 和 BAUDCON<3>) 的设置值产生两种不同的波特率时钟, 频率分别为移位速率的 16 倍或 64 倍。硬件不支持奇偶校验, 但可以用软件实现, 校验位保存在第 9 个数据位中。

当工作在异步模式下时, EUSART 模块包括以下重要组成部分:

- 波特率发生器
- 采样电路
- 异步发送器
- 异步接收器
- 同步间隔字符自动唤醒
- 12 位间隔字符发送
- 自动波特率检测

16.2.1 EUSART 异步发送器

图 16-3 显示了 EUSART 发送器的框图。发送器的核心是发送 (串行) 移位寄存器 (Transmit Shift Register, TSR)。移位寄存器从读 / 写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器中的数据由软件写入。在前一次装入的停止位发送完成前, 不会向 TSR 寄存器装入数据。一旦停止位发送完毕, TXREG 寄存器中的新数据 (如果有的话) 就会被装入 TSR。

一旦 TXREG 寄存器向 TSR 寄存器传输了数据 (在 1 个 Tcy 内发生), TXREG 寄存器就为空, 同时标志位 TXIF (PIR1<4>) 置 1。可以通过将中断允许位 TXIE (PIE1<4>) 置 1 或清零来允许 / 禁止该中断。不管 TXIE 的状态如何,

只要中断发生, TXIF 就会置 1 并且不能由软件清零。TXIF 不会在 TXREG 装入新数据时立即被清零, 而是在装入指令后的第二个指令周期被清零。因此在 TXREG 装入新数据后立即查询 TXIF, 会返回无效结果。

TXIF 表示的是 TXREG 寄存器的状态, 而另一个位 TRMT (TXSTA<1>) 则表示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要确定 TSR 寄存器是否为空, 用户只能对此位进行查询。

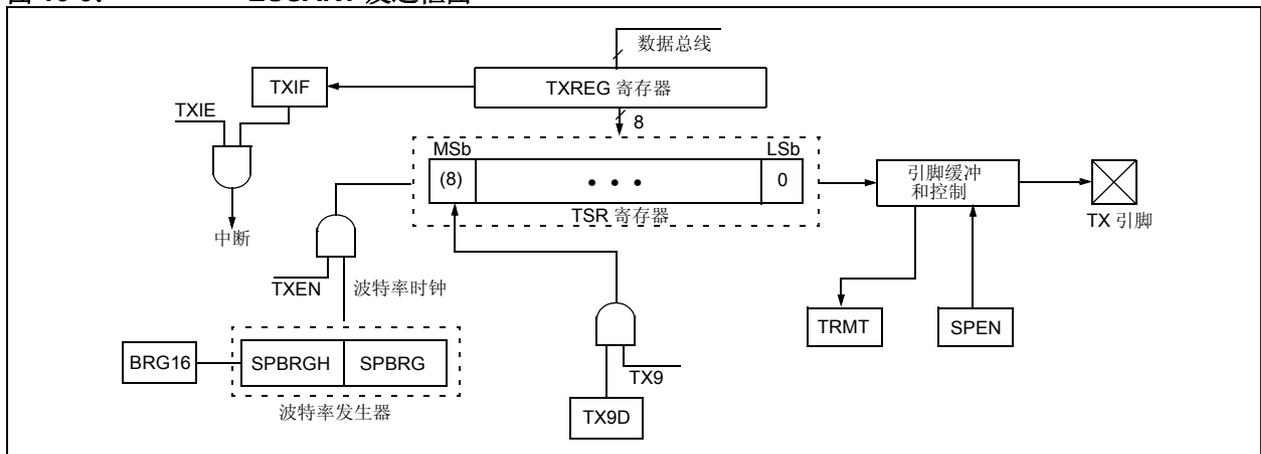
注 1: TSR 寄存器并未映射到数据存储寄存器中, 因此用户不能直接访问它。

2: 当使能位 TXEN 置 1 时, 标志位 TXIF 也置 1。

设置异步发送操作的步骤如下:

1. 对 SPBRGH:SPBRG 寄存器进行初始化, 以设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零, 以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串行端口。
3. 若需要中断, 将中断允许位 TXIE 置 1。
4. 若需要发送 9 位数据, 将发送位 TX9 置 1。发送的第 9 位可以是地址位也可以是数据位。
5. 通过将 TXEN 位置 1 使能发送, 此操作同时也会将 TXIF 位置 1。
6. 如果选择发送 9 位数据, 应该将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG 寄存器 (启动发送)。
8. 若想使用中断, 确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 16-3: EUSART 发送框图



PIC18F45J10 系列

图 16-4: 异步发送时序

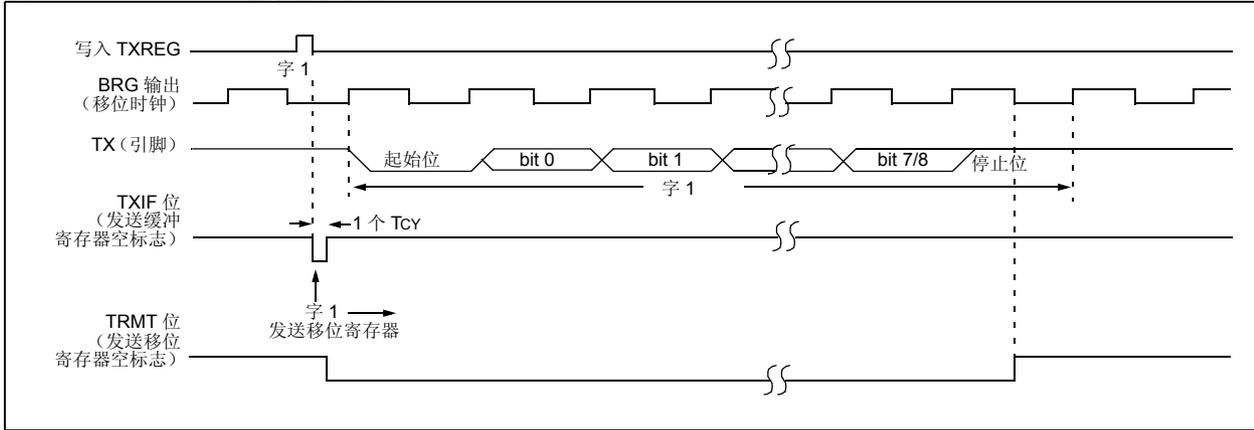


图 16-5: 异步发送时序 (背对背)

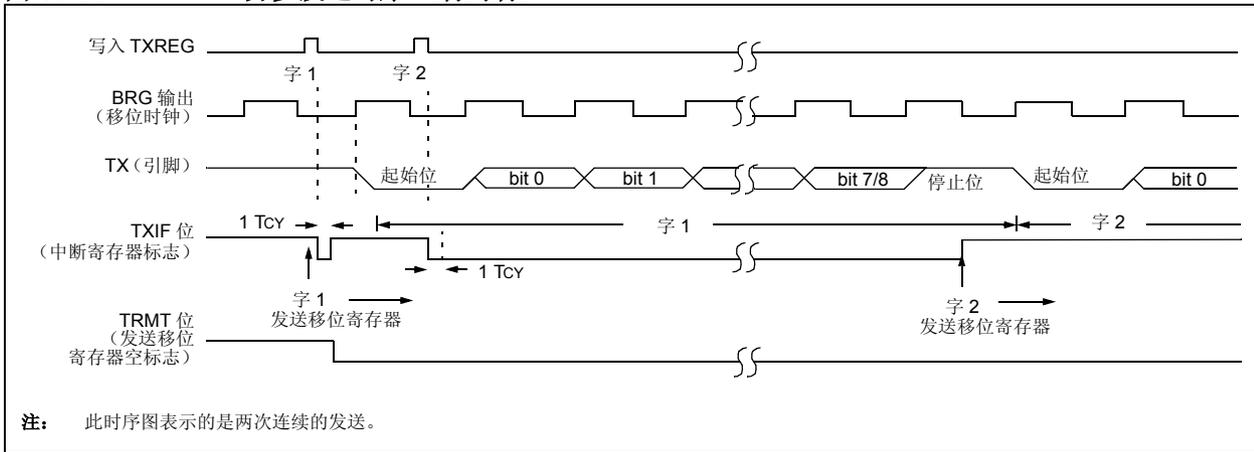


表 16-5: 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
TXREG	EUSART 发送寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注：— = 未用 (读为 0)。异步发送不使用阴影单元。

注 1: 在 28 引脚器件中这些位不被使用，并且应该读为 0。

16.2.2 EUSART 异步接收器

图 16-6 显示了接收器的框图。在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而主接收串行移位器的工作频率等于比特率或 Fosc。此模式通常用于 RS-232 系统。

设置异步接收操作的步骤如下：

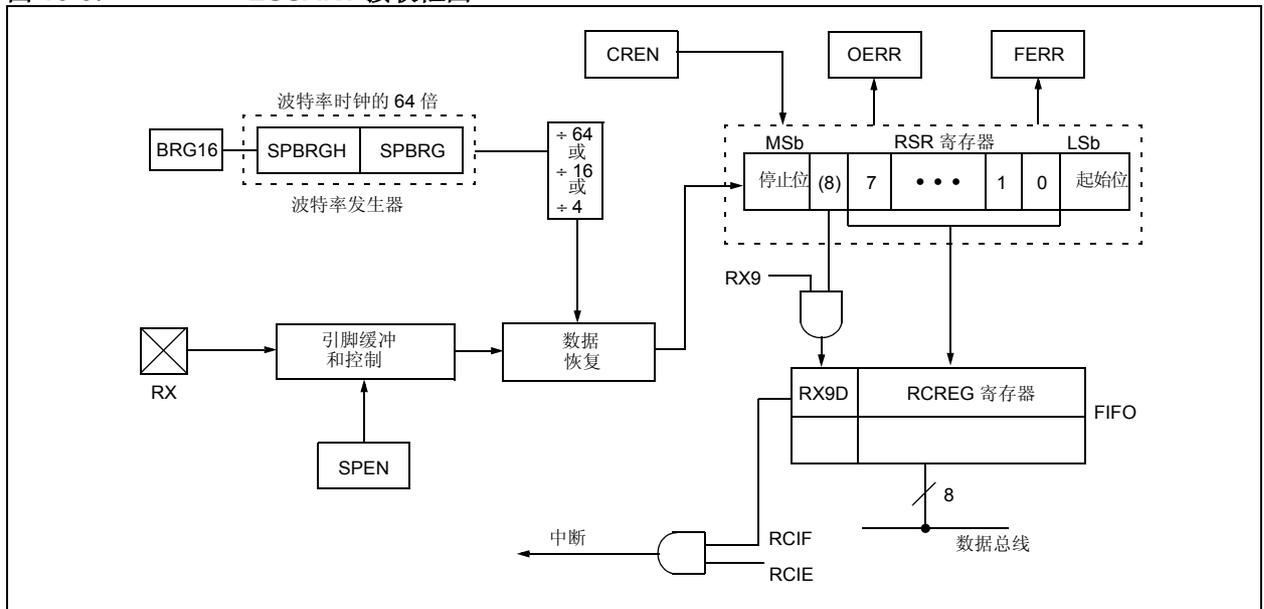
1. 对 SPBRGH:SPBRG 寄存器进行初始化，以设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零、SPEN 位置 1，使能异步串行端口。
3. 如果需要中断，将中断允许位 RCIE 置 1。
4. 若需要接收 9 位数据，将 RX9 位置 1。
5. 通过将 CREN 位置 1，使能接收。
6. 当接收完成时标志位 RCIF 将置 1，此时如果中断允许位 RCIE 已置 1，还将产生一个中断。
7. 读 RCSTA 寄存器以获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
8. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
9. 如果发生错误，通过将使能位 CREN 清零以清除错误。
10. 若想使用中断，确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

16.2.3 设置带有地址检测功能的 9 位模式

此模式通常用于 RS-485 系统。按如下步骤设置带有地址检测功能的异步接收操作：

1. 对 SPBRGH:SPBRG 寄存器进行初始化，以设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标的波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1，使能异步串行端口。
3. 若需要中断，将 RCEN 位置 1 并使用 RCIP 位设置优先级。
4. 将 RX9 位置 1，使能 9 位接收。
5. 将 ADDEN 位置 1，使能地址检测。
6. 将 CREN 位置 1，使能接收。
7. 当接收完成时 RCIF 位将置 1。此时如果 RCIE 和 GIE 位已置 1，还将响应中断。
8. 读 RCSTA 寄存器判断在接收时是否发生了错误，同时读取第 9 位数据（如果适用）。
9. 读 RCREG 来判断是否正在对器件进行寻址。
10. 如果发生错误，将 CREN 位清零。
11. 如果已经找到器件，将 ADDEN 位清零，允许接收到的数据进入接收缓冲器，并中断 CPU。

图 16-6: EUSART 接收框图



PIC18F45J10 系列

图 16-7: 异步接收时序

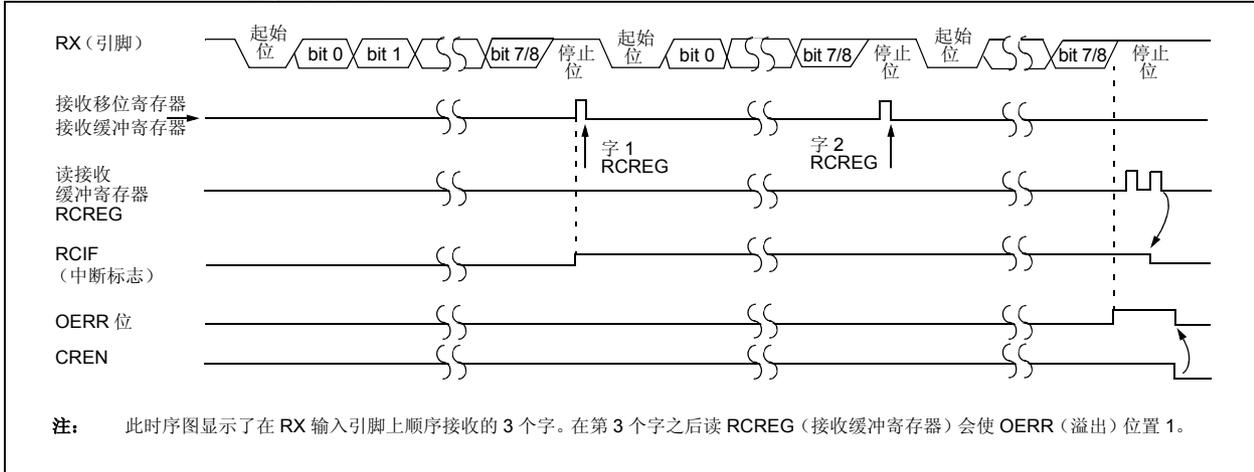


表 16-6: 与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
RCREG	EUSART 接收寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注： — = 未用（读为 0）。异步接收不使用阴影单元。

注 1： 在 28 引脚器件中这些位不被使用，并且应该读为 0。

16.2.4 同步间隔字符自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于非激活状态，并且无法进行正确的字节接收。自动唤醒功能允许在 RX/DT 线上有事件发生时唤醒控制器，该功能需要 EUSART 工作在异步模式下。

通过将 WUE 位（BAUDCON<1>）置 1，使能自动唤醒功能。该功能启用后，将禁止 RX/DT 上的典型接收操作，且 EUSART 保持在空闲状态并监视唤醒事件（与 CPU 运行模式无关）。唤醒事件是指 RX/DT 线上发生高电平到低电平的转换。（这与“同步间隔”字符或 LIN 协议唤醒信号字符的启动条件一致。）

唤醒事件后，模块会产生一个 RCIF 中断。在正常工作模式下，中断会与 Q 时钟同步产生（图 16-8）；如果器件处于休眠模式，则两者不同步（图 16-9）。通过读 RCREG 寄存器可清除中断条件。

唤醒事件后，当 RX 线上出现由低向高的电平转换时，WUE 位自动清零。此时，EUSART 模块将从空闲状态返回正常工作模式。由此用户可知“同步间隔”事件已经结束。

16.2.4.1 使用自动唤醒功能的注意事项

因为自动唤醒功能是通过检测RX/DT上的上升沿跳变实现的，所以在停止位前该引脚上任何的状态改变都可能产生错误的结束信号并导致数据或帧错误。因此，为了确保正确的传输，必须首先发送全0字符。对于标准的RS-232器件，这可以是00h（8位），而对于LIN总线则是000h（12位）。

另外还必须考虑振荡器起振时间，尤其在采用起振延迟较长的振荡器（即，HS模式）应用中更要注意这一点。“同步间隔”（或唤醒信号）字符必须足够长，并且跟有足够长的时间间隔，以便使选定振荡器有充足的时间起振并保证EUSART正确初始化。

16.2.4.2 使用WUE位的注意事项

用WUE和RCIF事件的时序来判断接收数据的有效性可能会引起混淆。如前所述，将WUE位置1会使EUSART进入空闲模式。唤醒事件会产生一个接收中断，并将RCIF位置1。此后当RX/DT出现上升沿时WUE位被清零。然后通过读RCREG寄存器清除中断条件。一般情况下，RCREG中的数据是无效数据，应该丢弃。

WUE位清零（或仍然置1）且RCIF标志位置1并不能表明RCREG中数据接收是完整的。用户应该考虑使用固件验证是否完整地接收了数据。

要确保没有丢失有效数据，应检查RCIDL位来验证是否还在接收数据。如果不在进行接收，则可将WUE位置1，使器件立即进入休眠模式。

图 16-8: 正常工作模式下的自动唤醒位 (WUE) 时序

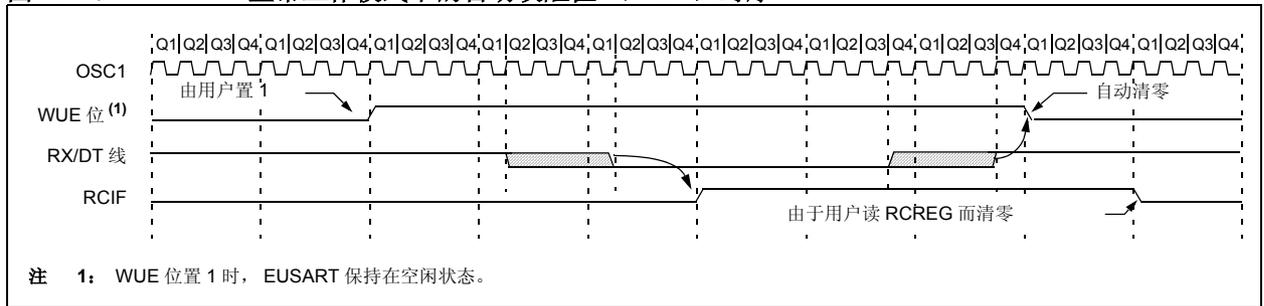
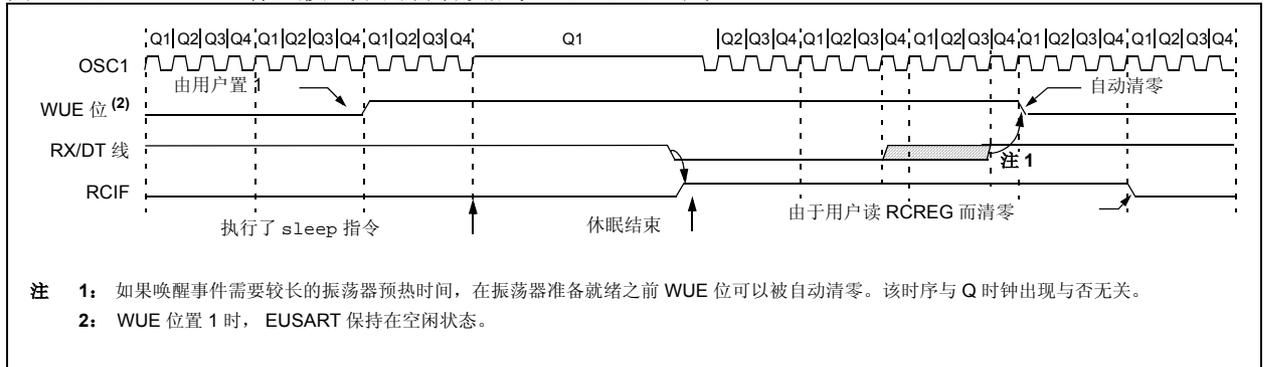


图 16-9: 休眠模式下的自动唤醒位 (WUE) 时序



PIC18F45J10 系列

16.2.5 间隔字符时序

EUSART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。发送的间隔字符包括 1 个起始位，后面跟有 12 个 0 位和一个停止位。当发送移位寄存器装有数据时，只要 SENDB 和 TXEN 位 (TXSTA<3> 和 TXSTA<5>) 置 1，就会发送帧间隔字符。请注意写入 TXREG 的数据值会被忽略，并会发送全 0。

在发送了相应的停止位后，硬件会自动将 SENDB 位清零。这样用户可以在发送完间隔字符（在 LIN 规范中通常是同步字符）后将下一个要发送的字节预先装入发送 FIFO。

请注意收到间隔字符时写入 TXREG 的数据值会被忽略。写入仅仅是为了启动正确的时序。

正如其在正常发送操作中一样，TRMT 位表明发送正在进行还是处于空闲状态。关于发送间隔字符时序，请参见图 16-10。

16.2.5.1 间隔和同步发送序列

以下序列会发送一个报文帧头，包括一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线主器件。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1，以设置间隔字符。
3. 将无效字符装入 TXREG，启动发送（该值会被忽略）。
4. 将 55h 写入 TXREG，以便把同步字符装入发送 FIFO 缓冲器。
5. 间隔字符发送后，硬件会将 SENDB 位复位。此时，同步字符会以预先配置的模式发送。

当 TXREG 为空时（由 TXIF 反映），下一个数据字节会写入 TXREG。

16.2.6 接收间隔字符

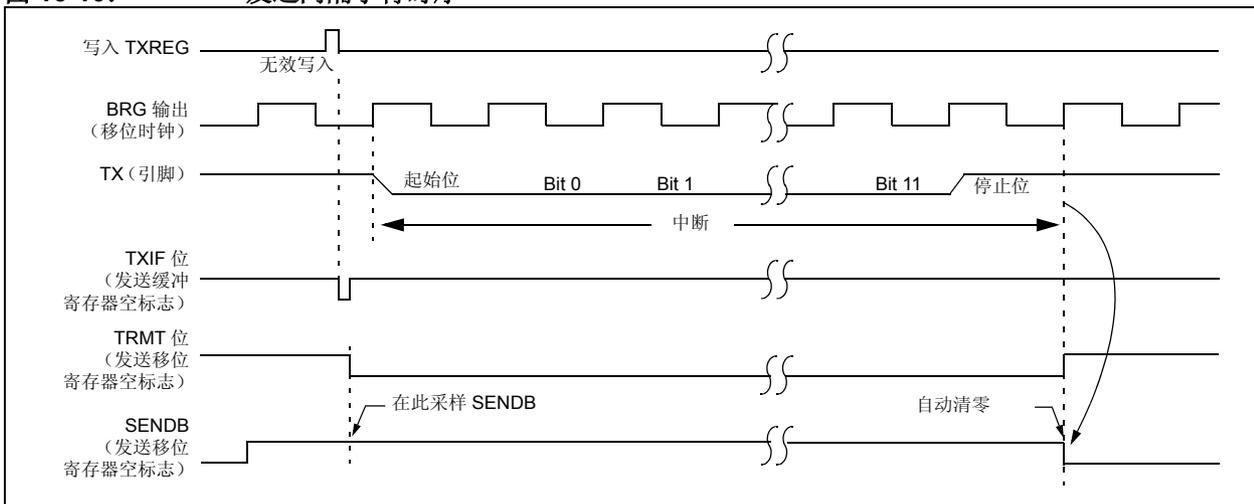
增强型 USART 模块接收间隔字符有两种方法。

第一种方法是强制将波特率配置为典型速率的 9/13。这可以使停止位在正确的采样点（对于间隔字符为起始位之后的 13 位，对于典型数据则是 8 个数据位）产生。

第二种方法使用第 16.2.4 节“同步间隔字符自动唤醒”中描述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX/DT 上电平的下两个跳变，产生一个 RCIF 中断，接收下一个数据字节，之后产生另一个中断。

请注意在间隔字符后，用户通常希望使能自动波特率检测功能。无论使用哪种方法，用户都可以在检测到 TXIF 中断时马上将 ABD 位置 1。

图 16-10: 发送间隔字符时序



16.3 EUSART 同步主控模式

将 CSRC 位 (TXSTA<7>) 置 1 可以进入同步主控模式。在此模式中, 数据以半双工方式 (即发送和接收不同时进行) 发送。发送数据时, 禁止接收, 反之亦然。将 SYNC 位 (TXSTA<4>) 置 1 可进入同步模式。此外, 应将使能位 SPEN (RCSTA<7>) 置 1, 分别把 TX 和 RX 引脚配置为 CK (时钟) 和 DT (数据) 线。

主控模式意味着处理器在 CK 时钟线上发送主控时钟信号。时钟极性是通过 SCKP 位 (BAUDCON<4>) 选择的。将 SCKP 置 1 是将空闲状态时的 CK 设为高电平, 将该位清零则将空闲状态时的 CK 设为低电平。此选项支持将本模块与 Microwire 器件配合使用。

16.3.1 EUSART 同步主控发送

图 16-3 显示了 EUSART 发送器的框图。发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后, 才向 TSR 寄存器装入新数据。一旦最后一位发送完成, 就会将 TXREG 寄存器中的新数据 (如果有的话) 装入 TSR。

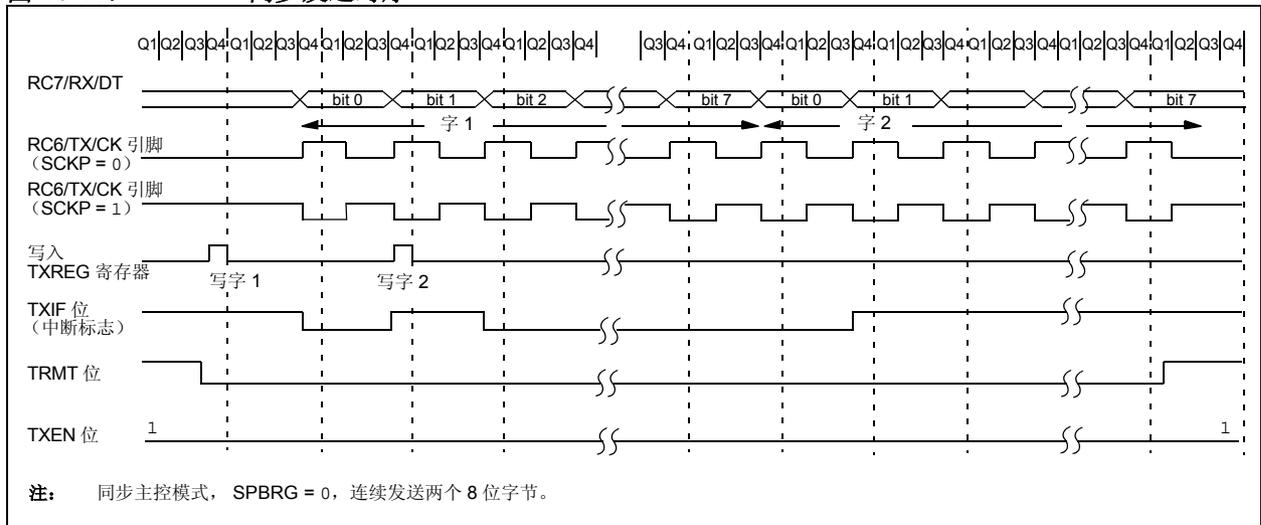
一旦 TXREG 寄存器向 TSR 寄存器传输了数据 (在 1 个 T_{cY} 内发生), TXREG 寄存器就为空, 同时标志位 TXIF (PIR1<4>) 置 1。可以通过将中断允许位 TXIE (PIE1<4>) 置 1 或清零来允许或禁止该中断。TXIF 的设置不受 TXIE 状态的影响, 且不能用软件清零。只有在新数据写入 TXREG 寄存器后, TXIF 位才会复位。

TXIF 表示的是 TXREG 寄存器的状态, 而另一个标志位 TRMT (TXSTA<1>) 则表示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要判断 TSR 寄存器是否为空, 用户只能对此位进行查询。TSR 并未映射到数据存储寄存器中, 因此用户不能直接访问它。

设置同步主控发送操作的步骤如下:

1. 对 SPBRGH:SPBRG 寄存器进行初始化, 以设置合适的波特率。按需要将 BRG16 位置 1 或清零, 以获得目标波特率。
2. 将 SYNC、SPEN 和 CSRC 位置 1, 使能同步主控串行端口。
3. 若需要中断, 将中断允许位 TXIE 置位。
4. 若需要发送 9 位数据, 将发送位 TX9 置 1。
5. 将 TXEN 位置 1, 使能发送。
6. 如果选择发送 9 位数据, 将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG 寄存器, 启动发送。
8. 若想使用中断, 确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 16-11: 同步发送时序



PIC18F45J10 系列

图 16-12: 同步发送时序 (由 TXEN 控制)

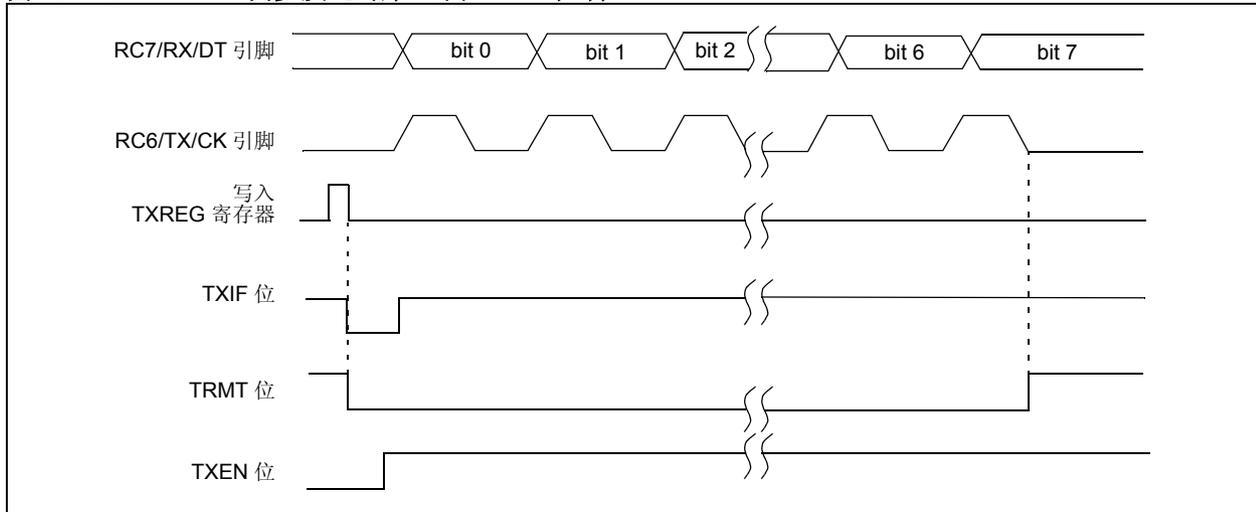


表 16-7: 与同步主控发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
TXREG	EUSART 发送寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注: — = 未用 (读为 0)。同步主控发送不使用阴影单元。

注 1: 在 28 引脚器件中这些位不被使用, 并且应该读为 0。

16.3.2 EUSART 同步主控接收

一旦选择了同步模式，只要将单字节接收使能位 SREN (RCSTA<5>) 或连续接收使能位 CREN (RCSTA<4>) 置 1，即可使能接收。在时钟的下降沿采样 RX 引脚上的数据。

如果将使能位 SREN 置 1，则只接收单个字。如果将使能位 CREN 置 1，则会连续接收数据，直到将 CREN 位清零。如果两个位均被置 1，则 CREN 具有优先权。

设置同步主控接收操作的步骤如下：

1. 对 SPBRGH:SPBRG 寄存器进行初始化，以设置合适的波特率。按需要将 BRG16 位置 1 或清零，以获得目标波特率。
2. 将 SYNC、SPEN 和 CSRC 位置 1，使能同步主控串行端口。

3. 确保将 CREN 和 SREN 位清零。
4. 如果需要中断，将中断允许位 RCIE 置 1。
5. 若需要接收 9 位数据，将 RX9 位置 1。
6. 若需要单字节接收，将 SREN 位置 1；若需要连续接收，将 CREN 位置 1。
7. 当接收完成时，中断标志位 RCIF 将置 1，此时如果中断允许位 RCIE 已置 1，还将产生一个中断。
8. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
9. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
10. 如果发生错误，将 CREN 位清零以清除错误。
11. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 16-13: 主控模式同步接收的时序 (由 SREN 位控制)

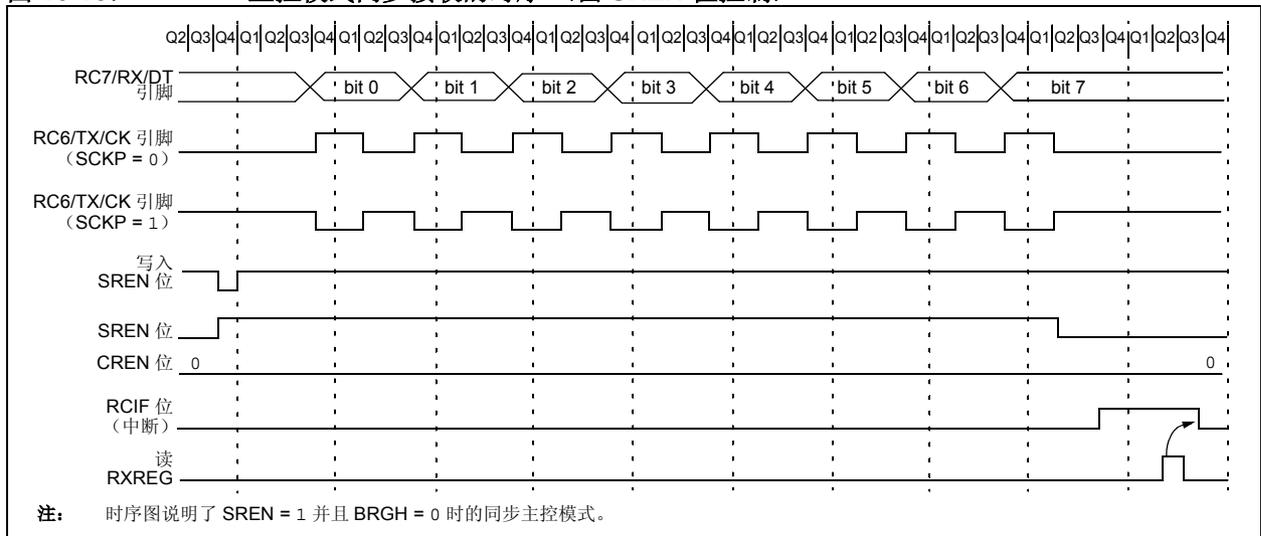


表 16-8: 与同步主控接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
RCREG	EUSART 接收寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注：— = 未用（读为 0）。同步主控接收不使用阴影单元。

注 1：在 28 引脚器件中这些位不被使用，并且应该读为 0。

PIC18F45J10 系列

16.4 EUSART 同步从动模式

将 CSRC (TXSTA<7>) 位清零可进入同步从动模式。此模式与同步主控模式的区别在于移位时钟由 CK 引脚上的外部时钟提供 (主控模式中由内部时钟提供)。这使得器件能在任何低功耗模式下发送或接收数据。

16.4.1 EUSART 同步从动发送

除了休眠模式以外, 同步主控和从动模式的工作方式是完全相同的。

如果向 TXREG 写入两个字, 然后执行 SLEEP 指令, 则将发生以下事件:

- 第一个字立即传送到 TSR 寄存器进行发送。
- 第二个字仍保留在 TXREG 寄存器中。
- 不会将标志位 TXIF 置 1。
- 当第一个字移出 TSR 后, TXREG 寄存器将把第二个字传送给 TSR, 同时将标志位 TXIF 置 1。
- 如果中断允许位 TXIE 已置 1, 中断将把器件从休眠状态唤醒。如果允许了全局中断, 程序则会跳转到中断矢量处执行。

设置同步从动发送的步骤如下:

- 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零, 使能同步从动串行端口。
- 将 CREN 和 SREN 位清零。
- 若需要中断, 将中断允许位 TXIE 置 1。
- 若需要发送 9 位数据, 将 TX9 位置 1。
- 将使能位 TXEN 置 1 使能发送。
- 如果选择发送 9 位数据, 将第 9 位数据装入 TX9D 位。
- 将数据装入 TXREG 寄存器, 启动发送。
- 若想使用中断, 请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

表 16-9: 与同步从动发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
TXREG	EUSART 发送寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注: — = 未用 (读为 0)。同步从动发送不使用阴影单元。

注 1: 在 28 引脚器件中这些位不被使用, 并且应该读为 0。

16.4.2 EUSART 同步从动接收

除了休眠模式、空闲模式以及在从动模式下忽略 SREN 位以外，同步主控和同步从动模式的工作方式完全相同。

如果在进入休眠或空闲模式前将 CREN 位置 1，使能接收，那么在该低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到 RCREG 寄存器，如果中断允许位 RCIE 已置 1，产生的中断将把器件从低功耗模式唤醒。如果允许了全局中断，程序则会跳转到中断矢量处执行。

设置同步从动接收操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步主控串行端口。
2. 若需要中断，将中断允许位 RCIE 置 1。
3. 若需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1，使能接收。
5. 当接收完成时，标志位 RCIF 将置 1。如果中断允许位 RCIE 已置 1，还将产生中断。
6. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
7. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，将 CREN 位清零以清除错误。
9. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

表 16-10: 与同步从动接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	45
RCREG	EUSART 接收寄存器								45
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	45
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	45
SPBRGH	EUSART 波特率发生器寄存器的高字节								45
SPBRG	EUSART 波特率发生器寄存器的低字节								45

图注： — = 未用（读为 0）。同步从动接收不使用阴影单元。

注 1： 在 28 引脚器件中这些位不被使用，并且应该读为 0。

PIC18F45J10 系列

注:

17.0 10 位模数转换器 (A/D) 模块

28 引脚器件的模数 (A/D) 转换器模块具有 10 路输入，而 40/44 引脚器件的模数转换器则具有 13 路输入。此模块能将一个模拟输入信号转换成相应的 10 位数字信号。

此模块有 5 个寄存器：

- A/D 结果高位寄存器 (ADRESH)
- A/D 结果低位寄存器 (ADRESL)
- A/D 控制寄存器 0 (ADCON0)
- A/D 控制寄存器 1 (ADCON1)
- A/D 控制寄存器 2 (ADCON2)

如寄存器 17-1 所示，A/D 模块的工作由 ADCON0 寄存器控制。如寄存器 17-2 所示，端口引脚的功能由 ADCON1 寄存器配置。如寄存器 17-3 所示，由 ADCON2 寄存器配置 A/D 时钟源，可编程的采集时间和输出结果的对齐方式。

寄存器 17-1: **ADCON0: A/D 控制寄存器 0**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

bit 7 **ADCAL:** A/D 校准位
 1 = 在下一次 A/D 转换时进行校准
 0 = 正常的 A/D 转换器操作

bit 6 **未用:** 读为 0

bit 5-2 **CHS3:CHS0:** 模拟通道选择位
 0000 = 通道 0 (AN0)
 0001 = 通道 1 (AN1)
 0010 = 通道 2 (AN2)
 0011 = 通道 3 (AN3)
 0100 = 通道 4 (AN4)
 0101 = 通道 5 (AN5) (1,2)
 0110 = 通道 6 (AN6) (1,2)
 0111 = 通道 7 (AN7) (1,2)
 1000 = 通道 8 (AN8)
 1001 = 通道 9 (AN9)
 1010 = 通道 10 (AN10)
 1011 = 通道 11 (AN11)
 1100 = 通道 12 (AN12)
 1101 = 未用通道 (2)
 1110 = 未用通道 (2)
 1111 = 未用通道 (2)

注 1: 这些通道在 28 引脚器件上未用。

注 2: 在未用通道上执行转换会返回悬空输入的测量值。

bit 1 **GO/DONE:** A/D 转换状态位

当 ADON = 1 时:
 1 = A/D 转换正在进行
 0 = A/D 空闲

bit 0 **ADON:** A/D 模块使能位

1 = 使能 A/D 转换器模块
 0 = 禁止 A/D 转换器模块

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

寄存器 17-2:

ADCON1: A/D 控制寄存器 1

U-0 U-0 R/W-0 R/W-0 R/W-0 R/W (1) R/W (1) R/W (1)
(1)

—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
---	---	-------	-------	-------	-------	-------	-------

bit 7

bit 0

bit 7-6 未用: 读为 0

bit 5 **VCFG1:** 参考电压配置位 (VREF- 电压源)

1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0:** 参考电压配置位 (VREF+ 电压源)

1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG3:PCFG0:** A/D 端口配置控制位:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000 ⁽¹⁾	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111 ⁽¹⁾	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = 模拟输入

D = 数字 I/O

注 1: PCFG 位的 POR 值取决于 PBadEN 配置位的值。当 PBadEN = 1 时, PCFG<3:0> = 0000; 当 PBadEN = 0 时, PCFG<3:0> = 0111。

注 2: AN5 至 AN7 仅在 40/44 引脚器件上可用。

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

寄存器 17-3: ADCON2: A/D 控制寄存器 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	
bit 7								bit 0

bit 7 **ADFM:** A/D 结果格式选择位

1 = 右对齐
0 = 左对齐

bit 6 **未用:** 读为 0

bit 5-3 **ACQT2:ACQT0:** A/D 采集时间选择位

111 = 20 个 TAD
110 = 16 个 TAD
101 = 12 个 TAD
100 = 8 个 TAD
011 = 6 个 TAD
010 = 4 个 TAD
001 = 2 个 TAD
000 = 0 个 TAD⁽¹⁾

bit 2-0 **ADCS2:ADCS0:** A/D 转换时钟选择位

111 = FRC (由 A/D RC 振荡器产生时钟信号) ⁽¹⁾
110 = Fosc/64
101 = Fosc/16
100 = Fosc/4
011 = FRC (由 A/D RC 振荡器产生时钟信号) ⁽¹⁾
010 = Fosc/32
001 = Fosc/8
000 = Fosc/2

注 1: 如果选择了 A/D Frc 时钟源, 在 A/D 时钟启动之前会添加一个 Tcy (指令周期) 的延迟。这允许在开始转换之前执行 SLEEP 指令。

图注:

R = 可读位

W = 可写位

U = 未用位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

PIC18F45J10 系列

模拟参考电压可通过软件来选择，该参考电压可以是器件的正电源电压和负电源电压（VDD 和 VSS）或 RA3/AN3/VREF+ 引脚和 RA2/AN2/VREF-/CVREF 引脚上的电压。

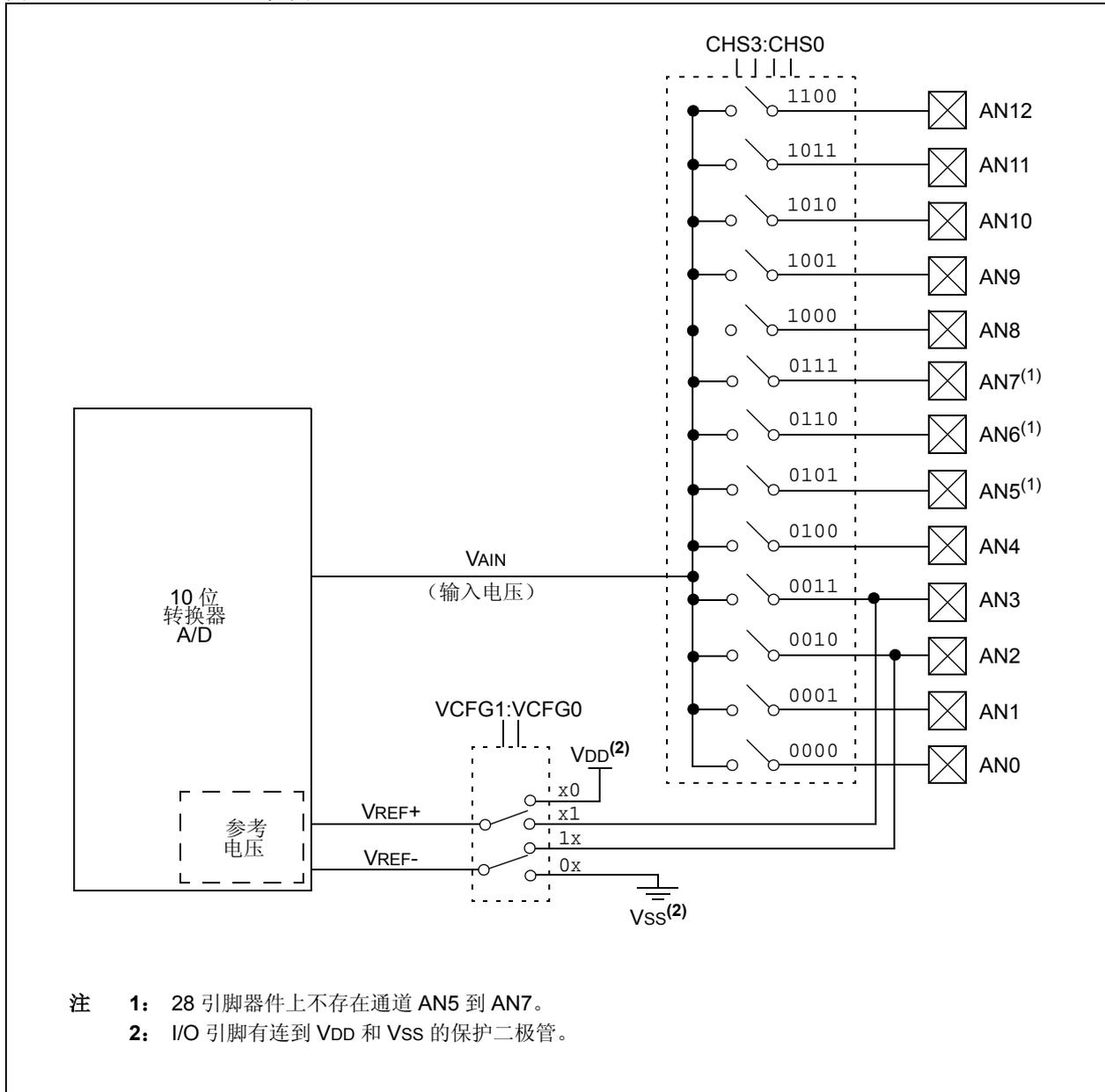
A/D 转换器具备一个独特的特性，它可在器件处于休眠模式时正常工作。要使 A/D 转换器在休眠模式下运行，A/D 转换时钟必须来自于 A/D 模块内部的 RC 振荡器。

采样保持电路的输出是转换器的输入，A/D 转换器采用逐次逼近法产生转换结果。

器件复位强制所有寄存器进入复位状态。同时迫使 A/D 模块关闭并中止任何正在进行的转换。

可以将每个与 A/D 转换器相关的端口引脚配置为模拟输入或数字 I/O。ADRESH 和 ADRESL 寄存器保存 A/D 转换的结果。当 A/D 转换完成之后，转换结果被载入 ADRESH:ADRESL 寄存器对，GO/DONE 位（ADCON0 寄存器）被清零且 A/D 中断标志位 ADIF 置 1。A/D 模块的框图见图 17-1。

图 17-1: A/D 框图



上电复位时，ADRESH:ADRESL 寄存器中的值保持不变。上电复位后，ADRESH:ADRESL 寄存器中的值不确定。

在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采集。必须将模拟输入通道相应的 TRIS 位选择为输入。采集时间的确定请参见第 17.1 节“A/D 采集要求”。在采集完成后，A/D 转换即可开始。可将采集时间编程设定为在 GO/DONE 位置 1 和实际转换启动之间。

在执行 A/D 转换时应该遵循以下步骤：

1. 配置 A/D 模块：
 - 配置模拟引脚、参考电压和数字 I/O（通过 ADCON1 寄存器）
 - 选择 A/D 输入通道（通过 ADCON0 寄存器）
 - 选择 A/D 采集时间（通过 ADCON2 寄存器）
 - 选择 A/D 转换时钟（通过 ADCON2 寄存器）
 - 打开 A/D 模块（通过 ADCON0 寄存器）
2. 需要时，配置 A/D 中断：
 - 将 ADIF 位清零
 - 将 ADIE 位置 1
 - 将 GIE 位置 1
3. 如果需要的话，等待所要求的采集时间。
4. 启动转换：
 - 将 GO/DONE 位（ADCON0 寄存器）置 1

5. 等待 A/D 转换完成，通过以下两种方法之一可判断转换是否完成：
 - 查询 GO/DONE 位是否被清零
 或
 - 等待 A/D 转换中断
6. 读取 A/D 结果寄存器（ADRESH:ADRESL）；需要时将 ADIF 位清零。
7. 如需再次进行 A/D 转换，请根据要求返回步骤 1 或步骤 2。将每位的 A/D 转换时间定义为 TAD。在下次采集开始前至少需要等待 2 个 TAD。

图 17-2: A/D 转换方式

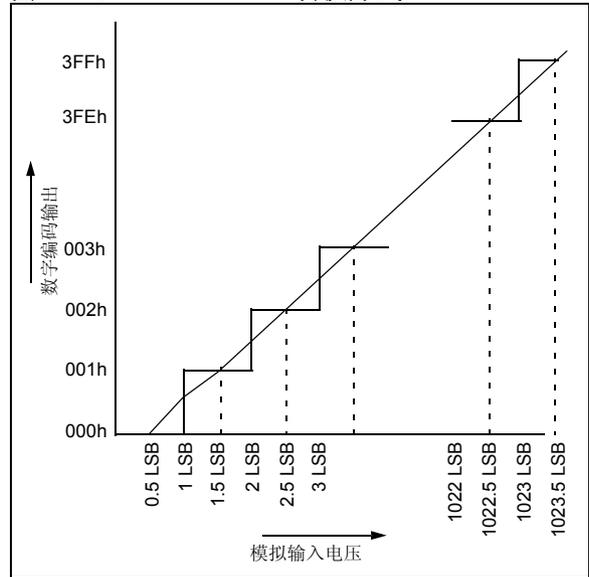
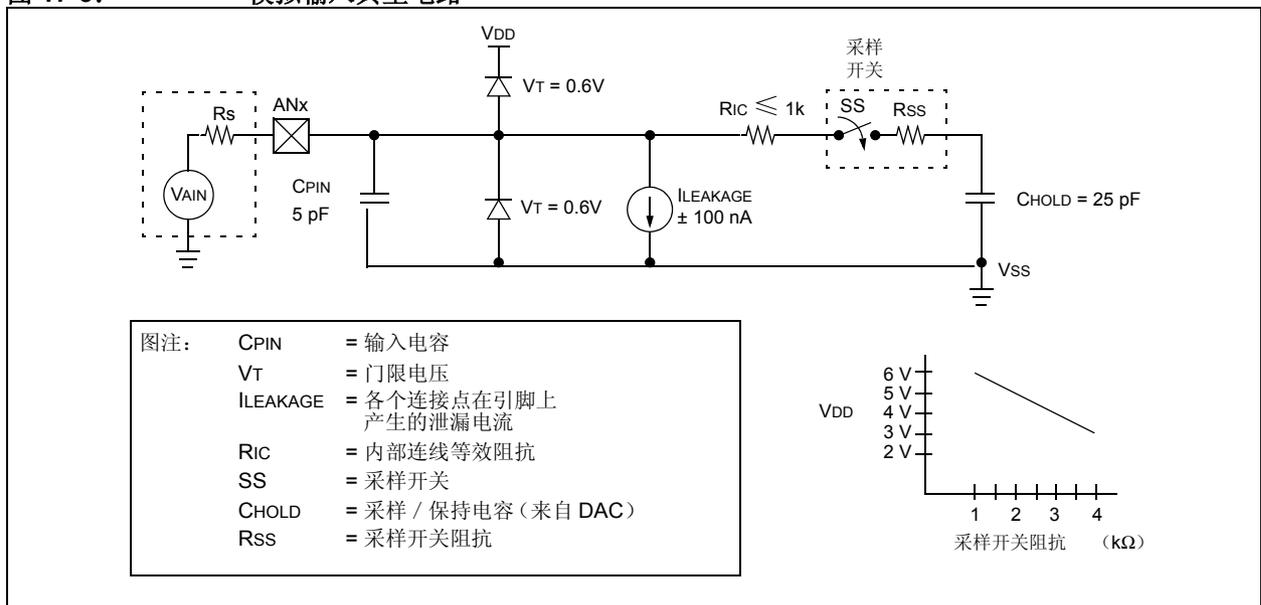


图 17-3: 模拟输入典型电路



PIC18F45J10 系列

17.1 A/D 采集要求

为了使 A/D 转换器达到规定精度，必须让充电保持电容 (CHOLD) 充满至输入通道的电平。图 17-3 显示了模拟输入的典型电路。源阻抗 (Rs) 和内部采样开关阻抗 (Rss) 直接影响给电容 CHOLD 充电所需要的时间。采样开关阻抗 (Rss) 随器件电压 (VDD) 变化而改变。源阻抗影响模拟输入的失调电压 (由于引脚泄漏电流)。建议模拟信号源的最大阻抗为 2.5 kΩ。在选择 (改变) 模拟输入通道后，必须对通道进行采样才能开始转换，采样时间必须大于最小采集时间。

注： 当开始转换时，应把保持电容从输入引脚断开。

可使用公式 17-1 来计算最小采集时间。该公式假设使用的量化误差为 1/2 LSB (A/D 转换需要 1024 步)。1/2 LSB 的误差是 A/D 模块达到规定分辨率所能允许的最大误差。

公式 17-3 显示了所需的最小采集时间 TACQ 的计算过程。计算结果是基于以下对应用系统的假设得出的：

CHOLD	=	25 pF
Rs	=	2.5 kΩ
转换误差	≤	1/2 LSB
VDD	=	5V → Rss = 2 kΩ
温度	=	85°C (系统最大值)

公式 17-1: 采集时间

$$\begin{aligned} TACQ &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= TAMP + TC + TCOFF \end{aligned}$$

公式 17-2: A/D 最小充电时间

$$\begin{aligned} V_{HOLD} &= (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{-(Tc/CHOLD)(R_{IC} + R_{SS} + R_s)}) \\ \text{或} \\ Tc &= -(CHOLD)(R_{IC} + R_{SS} + R_s) \ln(1/2048) \end{aligned}$$

公式 17-3: 计算所需的最小采集时间

$$\begin{aligned} TACQ &= TAMP + Tc + TCOFF \\ TAMP &= 0.2 \mu s \\ TCOFF &= (Temp - 25^\circ C)(0.02 \mu s/^\circ C) \\ &= (85^\circ C - 25^\circ C)(0.02 \mu s/^\circ C) \\ &= 1.2 \mu s \end{aligned}$$

只有在温度 >25°C 时才需要温度系数。当温度低于 25°C 时，TCOFF = 0 μs。

$$\begin{aligned} Tc &= -(CHOLD)(R_{IC} + R_{SS} + R_s) \ln(1/2047) \mu s \\ &= -(25 \text{ pF})(1 \text{ kW} + 2 \text{ kW} + 2.5 \text{ kW}) \ln(0.0004883) \mu s \\ &= 1.05 \mu s \\ TACQ &= 0.2 \mu s + 1 \mu s + 1.2 \mu s \\ &= 2.4 \mu s \end{aligned}$$

17.2 选择和配置采集时间

ADCON2 寄存器允许用户选择当 $\overline{\text{GO/DONE}}$ 位置 1 后的采集时间。它还为用户提供了使用系统自定采集时间的选项。

可以使用 ACQT2:ACQT0 位 (ADCON2<5:3>) 设置采集时间, 采集时间的范围是 2 到 20 TAD。当 $\overline{\text{GO/DONE}}$ 位置 1 时, A/D 模块继续对输入进行采样, 采样时间为所选择的采集时间, 然后自动开始转换。因为采集时间已被编程, 就不需要在选择通道以后等待一个采集时间才置位 $\overline{\text{GO/DONE}}$ 位。

若 ACQT2:ACQT0 = 000, 则表示选择手动采集。当 $\overline{\text{GO/DONE}}$ 位置 1 时, 采样停止, 转换开始。用户需确保在选择所需要的输入通道和 $\overline{\text{GO/DONE}}$ 置 1 之间经过了必需的采集时间。此选项也是 ACQT2:ACQT0 位的默认复位状态, 并且与不提供可编程采集时间的器件兼容。

在这两种情况下, 当转换完成时, $\overline{\text{GO/DONE}}$ 位被清零、ADIF 标志位被置 1 且 A/D 再次开始对当前选择的通道进行采样。如果编程了采集时间, 那么将不会有标志显示采集时间是否结束和转换是否开始。

17.3 选择 A/D 转换时钟

每位的 A/D 转换时间被定义为 TAD。每完成一次 10 位 A/D 转换需要 11 个 TAD。可用软件选择 A/D 转换时钟源。TAD 有以下 7 种可能的选择:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- 内部 RC 振荡器

要进行正确的 A/D 转换, A/D 转换时钟 (TAD) 必须尽可能小, 但它必须大于最小 TAD (欲知更多信息, 请参见参数 130)。

表 17-1 显示了器件在不同的工作频率下和选择不同的 A/D 时钟源时得到的 TAD。

表 17-1: 不同器件工作频率下的 TAD

AD 时钟源 (TAD)		最高器件频率	
工作模式	ADCS2:ADCS0	PIC18F2XJ10/4XJ10	PIC18LF2XJ10/4XJ10 ⁽⁴⁾
2 TOSC	000	2.86 MHz	1.43 MHz
4 TOSC	100	5.71 MHz	2.86 MHz
8 TOSC	001	11.43 MHz	5.72 MHz
16 TOSC	101	22.86 MHz	11.43 MHz
32 TOSC	010	40.0 MHz	22.86 MHz
64 TOSC	110	40.0 MHz	40.0 MHz
RC ⁽³⁾	x11	1.00 MHz ⁽¹⁾	1.00 MHz ⁽²⁾

注 1: RC 时钟源的典型 TAD 时间为 1.2 μs 。

2: RC 时钟源的典型 TAD 时间为 2.5 μs 。

3: 当器件工作频率高于 1 MHz 时, 整个转换过程必须在休眠模式下进行, 否则 A/D 转换精度可能超出规范允许的范围。

4: 仅低功耗器件 (PIC18LF2XJ10/4XJ10)。

PIC18F45J10 系列

17.4 在功耗管理模式下的工作方式

在功耗管理模式中，自动采集时间和 A/D 转换时钟的选择在一定程度上可由时钟源和频率决定。

如果要在器件处于功耗管理模式时进行 A/D 转换，ADCON2 中的 ACQT2:ACQT0 和 ADCS2:ADCS0 位就应该根据要使用的功耗管理模式时钟进行更新。在进入该模式之后，可以开始 A/D 采集或转换。采集或转换开始以后，器件仍应继续使用相同的时钟源直到转换完成。

如果需要的话，在转换期间也可以将器件置于相应的功耗管理空闲模式。如果器件时钟频率小于 1 MHz，就应该选择 A/D RC 时钟源。

在休眠模式下工作需要选择 A/D FRC 时钟。如果将 ACQT2:ACQT0 位设置为“000”并启动转换，转换将延迟一个指令周期以允许执行 SLEEP 指令并进入休眠模式。在转换开始前，IDLEN 位 (OSCCON<7>) 必须已经被清零。

17.5 A/D 转换器校准

PIC18F45J10 系列器件中的 A/D 转换器包含自校准特性，它可以补偿模块中所产生的任何偏移。可通过将 ADCAL 位 (ADCON0<7>) 置 1 来启动自动校准。下一次 GO/DONE 位被置 1 时，此模块将执行“虚拟”转换（即不读取任何输入通道）并将转换结果存储在内部以补偿偏移。因此，后续的偏移都将得到补偿。

在校准过程中假设器件处于相对稳定工作状态。如果使用 A/D 校准，它应该在每次器件复位之后或在工作条件发生其他重要变化时执行。

17.6 配置模拟端口引脚

ADCON1、TRISA、TRISB 和 TRISE 寄存器都控制 A/D 端口引脚的操作。若希望端口引脚为模拟输入，则必须将相应的 TRIS 位置 1（输入）。如果 TRIS 位被清零（输出），则将对引脚的数字输出电平 (V_{OH} 或 V_{OL}) 进行相应的转换。

A/D 转换操作与 CHS3:CHS0 位以及 TRIS 位的状态无关。

- | | |
|----------|---|
| 注 | 1: 读取端口寄存器时，所有配置为模拟输入通道的引脚均读为 0（低电平）。配置为数字输入的引脚将对模拟输入信号进行转换。引脚上的模拟电平将被正确转换为数字电平。 |
| | 2: 定义为数字输入引脚上的模拟电平可能会导致数字输入缓冲器消耗的电流超出器件规范。 |
| | 3: 配置寄存器 3H 中的 PBADEN 位通过控制 ADCON1 中的 PCFG0 位如何复位，可以把 PORTB 引脚配置为复位时作为模拟或数字引脚。 |

17.7 A/D 转换

图 17-4 显示了在 $\overline{GO/DONE}$ 位置 1 且 ACQT2:ACQT0 位被清零后 A/D 转换器的工作状态。转换在下一条指令执行之后开始，以允许器件在转换开始之前进入休眠模式。

图 17-5 显示了在 $\overline{GO/DONE}$ 位置 1 且 ACQT2:ACQT0 位被设置为“010”（即在转换开始之前选择了 4 TAD 的采集时间）后 A/D 转换器的工作状态。

在转换期间将 $\overline{GO/DONE}$ 位清零将中止当前的 A/D 转换。不会用尚未完成的 A/D 转换结果更新 A/D 结果寄存器对。这意味着 ADRESH:ADRESL 寄存器对仍将保持上一次转换完成后的值（或上一次写入 ADRESH:ADRESL 寄存器的值）。

在 A/D 转换完成或停止以后，需要等待 2 个 TAD 才能开始下一次采集。等待结束后将自动开始对所选通道进行采集。

注： 不应在启动 A/D 模块的同一指令中将 $\overline{GO/DONE}$ 位置 1。

图 17-4: A/D 转换 TAD 周期 ($ACQT<2:0> = 000, TACQ = 0$)

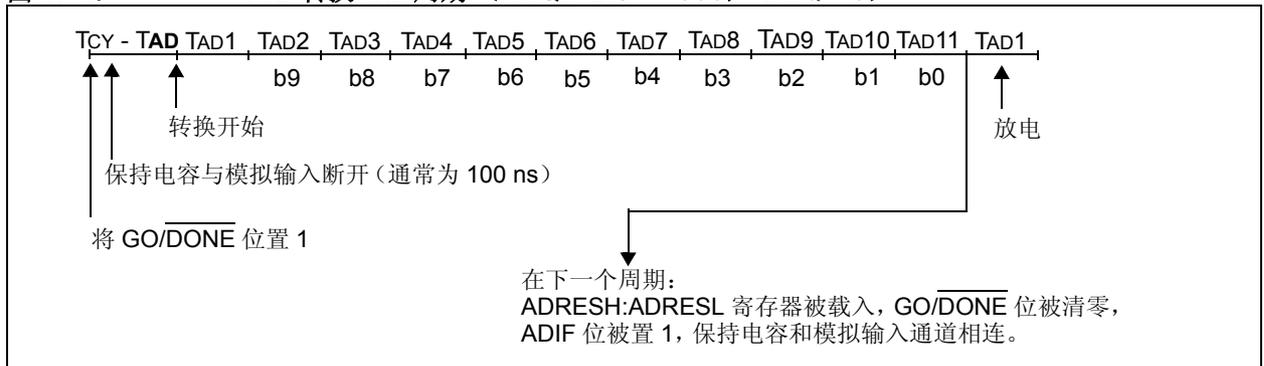
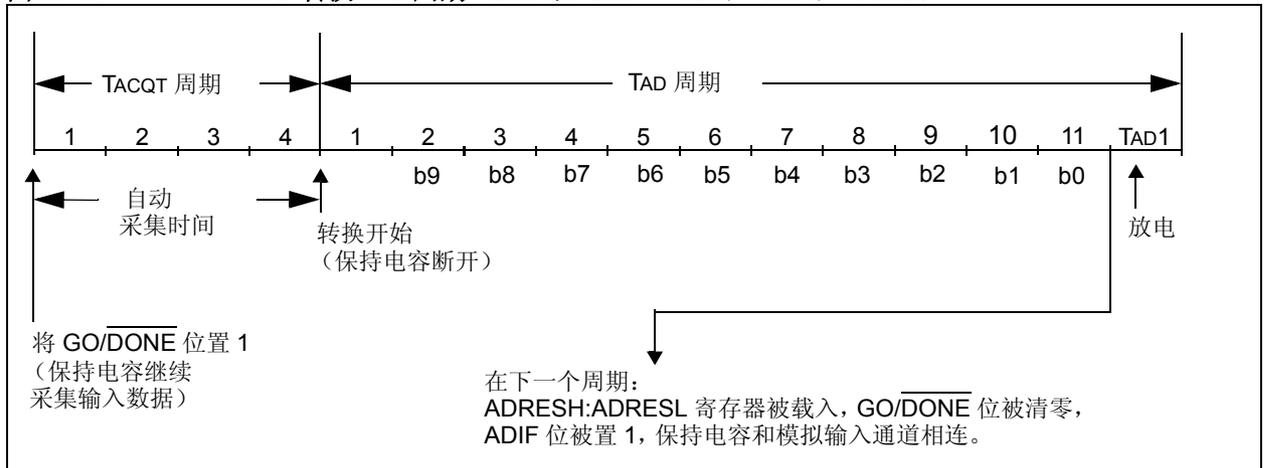


图 17-5: A/D 转换 TAD 周期 ($ACQT<2:0> = 010, TACQ = 4 TAD$)



PIC18F45J10 系列

17.8 使用 CCP2 触发信号

可以通过 CCP2 模块的“特殊事件触发信号”启动 A/D 转换。这需要将 CCP2M3:CCP2M0 位 (CCP2CON<3:0>) 编程为“1011”，且使能 A/D 模块 (ADON 位置 1)。发生触发事件时，GO/DONE 位被置 1，启动 A/D 采集和转换并将 Timer1 计数器复位为 0。复位 Timer1 可自动重复 A/D 采集周期，最大限度地降低了软件开销 (将 ADRESH:ADRESL 内容移到目标单元)。在“特殊事件触

发信号”将 GO/DONE 位置 1 (启动转换) 之前，用户必须选择正确的模拟输入通道并设定最小采集时间或者选择合适的 TAcq 时间。

如果未使能 A/D 模块 (ADON 清零)，则“特殊事件触发信号”将被 A/D 模块忽略，但它仍会将 Timer1 计数器复位。

表 17-2: 与 A/D 转换操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	43
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	45
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	45
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	45
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
ADRESH	A/D 结果寄存器高字节								44
ADRESL	A/D 结果寄存器低字节								44
ADCON0	ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	44
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	44
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	44
PORTA	—	—	RA5	—	RA3	RA2	RA1	RA0	46
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	46
TRISB	PORTB 数据方向控制寄存器								46
LATB	PORTB 数据锁存器 (读取和写入数据锁存器)								46
PORTE ⁽¹⁾	—	—	—	—	—	RE2	RE1	RE0	46
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	46
LATE ⁽¹⁾	—	—	—	—	—	PORTE 数据锁存器 (读取和写入数据锁存器)			46

图注: — = 未用, 读为 0。A/D 转换未使用阴影单元。

注 1: 这些寄存器或位在 28 引脚器件上不存在, 应该被读为 0。

18.0 比较器模块

模拟比较器模块包含两个比较器，可以用多种方式对它们进行配置。可以选择与 RA0 到 RA5 引脚复用的模拟输入及片上参考电压（见第 19.0 节“比较器参考电压模块”）作为输入。数字输出（正常或翻转的）可从引脚电平获取也可通过控制寄存器读取。

CMCON 寄存器（寄存器 18-1）选择比较器输入和输出配置。图 18-1 给出了各种比较器配置的框图。

寄存器 18-1: **CMCON: 比较器控制寄存器**

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7						bit 0	

- bit 7 **C2OUT**: 比较器 2 输出位
当 C2INV = 0 时:
 1 = C2 VIN+ > C2 VIN-
 0 = C2 VIN+ < C2 VIN-
当 C2INV = 1 时:
 1 = C2 VIN+ < C2 VIN-
 0 = C2 VIN+ > C2 VIN-
- bit 6 **C1OUT**: 比较器 1 输出位
当 C1INV = 0 时:
 1 = C1 VIN+ > C1 VIN-
 0 = C1 VIN+ < C1 VIN-
当 C1INV = 1 时:
 1 = C1 VIN+ < C1 VIN-
 0 = C1 VIN+ > C1 VIN-
- bit 5 **C2INV**: 比较器 2 输出翻转位
 1 = C2 输出翻转
 0 = C2 输出不翻转
- bit 4 **C1INV**: 比较器 1 输出翻转位
 1 = C1 输出翻转
 0 = C1 输出不翻转
- bit 3 **CIS**: 比较器输入开关位
当 CM2:CM0 = 110 时:
 1 = C1 VIN- 连接到 RA3/AN3/VREF+
 C2 VIN- 连接到 RA2/AN2/VREF-/CVREF
 0 = C1 VIN- 连接到 RA0/AN0
 C2 VIN- 连接到 RA1/AN1
- bit 2-0 **CM2:CM0**: 比较器模式位

图 18-1 给出了比较器的几种模式以及相应 CM2:CM1 位设置。

图注:

R = 可读位	W = 可写位	U = 未用位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

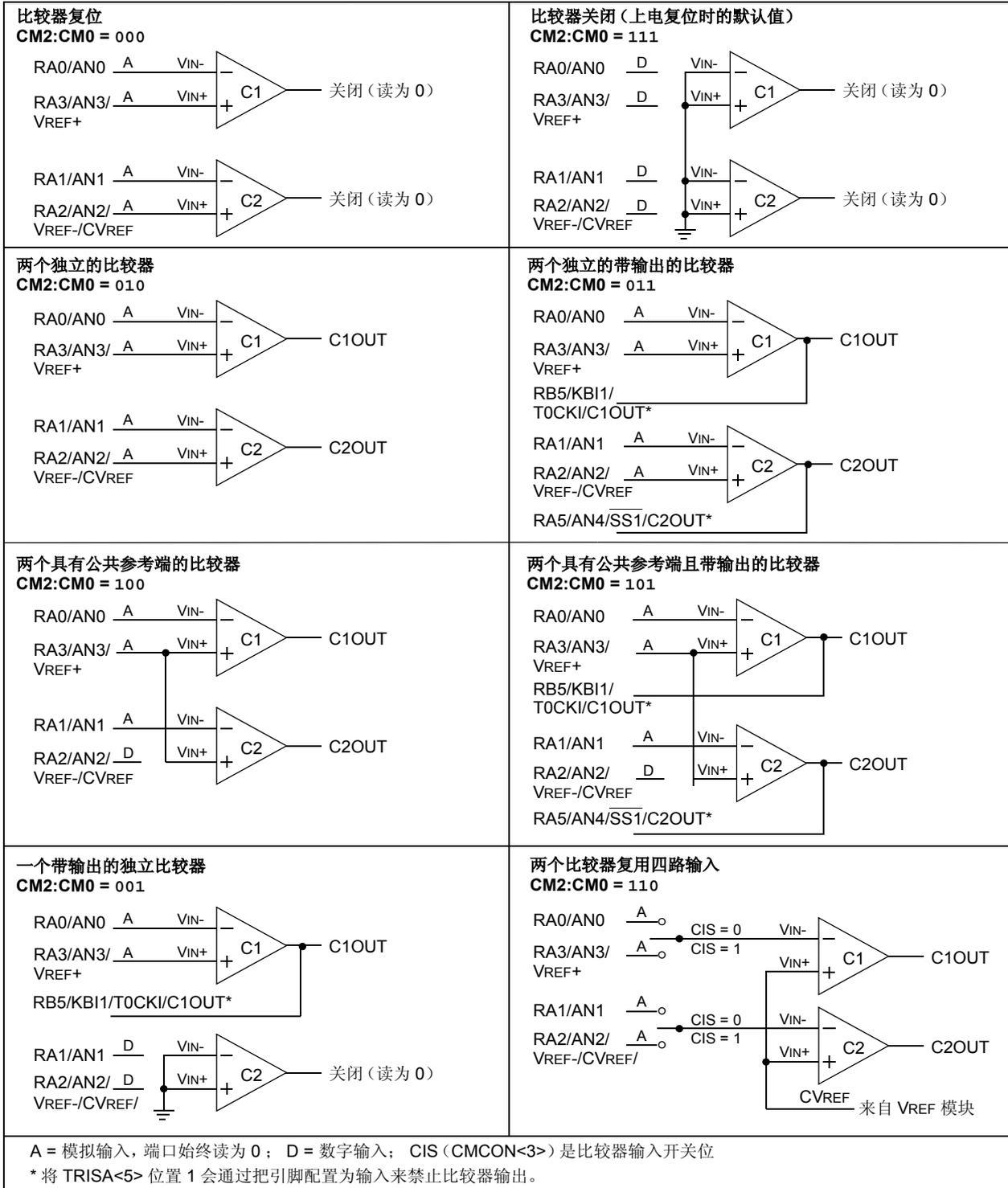
18.1 比较器配置

图 18-1 给出了比较器的 8 种工作模式。CMCON 寄存器的 CM2:CM0 位用于选择这些模式。TRISA 寄存器控制每种模式下比较器引脚的数据方向。如果改变比较器

模式，由于存在特定的模式改变延迟（如第 23.0 节“电气规范”所示）比较器的输出电平可能会在此延迟期间无效。

注： 改变比较器工作模式的过程中，应禁止比较器的中断，否则会产生错误中断。

图 18-1: 比较器 I/O 工作模式



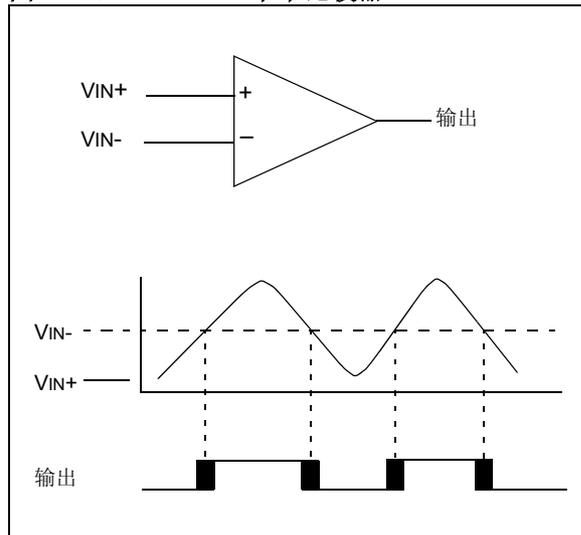
18.2 比较器工作原理

图 18-2 所示为单个比较器以及模拟输入电平和数字输出之间的关系。如果 V_{IN+} 上的模拟输入电平小于 V_{IN-} 上的模拟输入电平，那么比较器将输出数字低电平。当 V_{IN+} 上的模拟输入电平高于 V_{IN-} 上的模拟输入电平时，比较器输出数字高电平。图 18-2 中比较器输出的阴影部分表示因输入失调和响应时间所造成的不确定区域。

18.3 比较器参考电压

根据不同的工作模式，比较器可使用外部或内部参考电压。将加在 V_{IN-} 上的模拟信号和加在 V_{IN+} 上的信号相比较，并相应地调整比较器的数字输出（图 18-2）。

图 18-2: 单个比较器



18.3.1 外部参考信号

当使用外部参考电压时，可将比较器模块中的两个比较器配置为使用同一个参考源或使用不同的参考源。然而，门限检测器应用可能要求使用同一个参考源。参考信号必须在 V_{SS} 和 V_{DD} 之间，并且可被施加到比较器的任一引脚上。

18.3.2 内部参考信号

比较器模块也可以选择使用比较器参考电压模块内部产生的参考电压。第 19.0 节“比较器参考电压模块”中对此进行了更详细地说明。

只有在两个比较器复用四路输入的模式 ($CM2:CM0 = 110$) 下才可使用内部参考电压。该模式下，内部参考电压被施加到两个比较器的 V_{IN+} 引脚上。

18.4 比较器的响应时间

响应时间是指从选定一个新的参考电压或输入源到比较器输出达到一个有效电平的最短时间。如果内部参考电压发生了改变，在使用比较器的输出时必须考虑到内部参考电压的最大延时。否则，应该使用比较器的最大延时（见第 23.0 节“电气规范”）。

18.5 比较器输出

通过读 $CMCON$ 寄存器中的相应位可读取比较器的输出。这些位是只读的。比较器的输出也可以直接输出到 I/O 引脚 $RB5$ 和 $RA5$ 。当使能时， $RB5$ 和 $RA5$ 引脚输出路径上的多路开关将发生切换，并且各个引脚的输出将与比较器的输出异步。每个比较器的不确定区的大小与输入失调电压和响应时间有关，在电气规范里对此作出了说明。图 18-3 为比较器输出的框图。

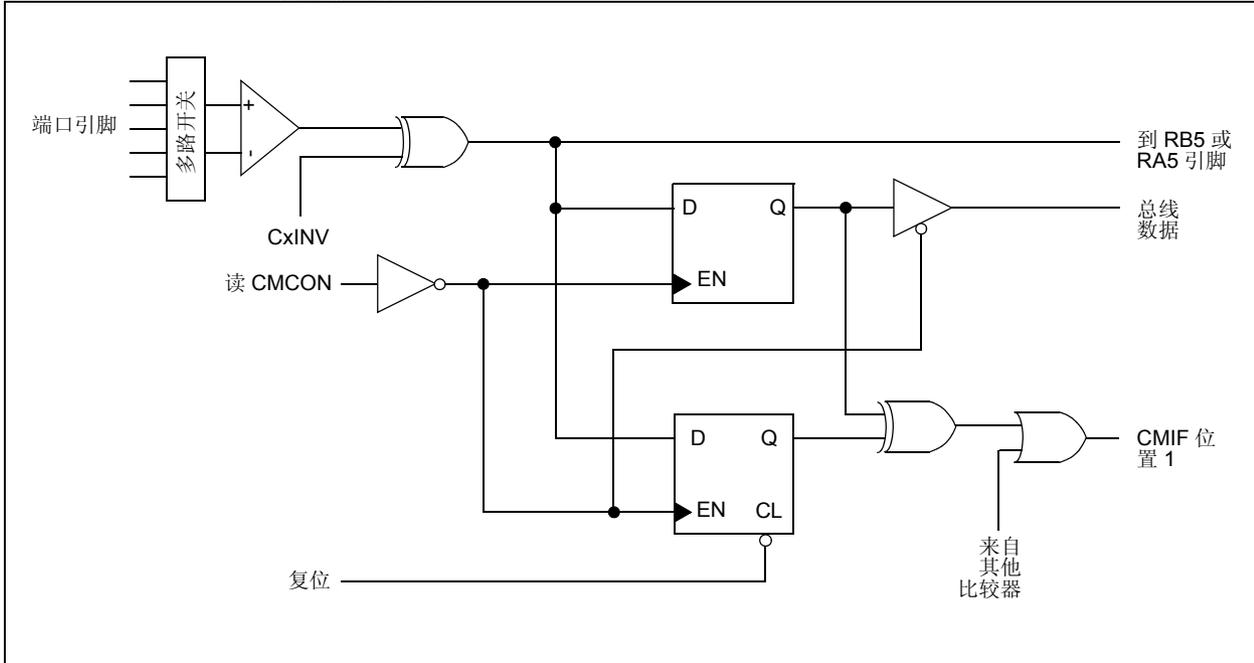
在该模式下， $TRISA$ 位仍作为 $RB5$ 和 $RA5$ 引脚的输出使能/禁止位。

使用 $C2INV$ 和 $C1INV$ 位 ($CMCON<5:4>$) 可以改变比较器输出的极性。

- 注 1:** 当读取端口寄存器时，所有配置为模拟输入的引脚都读为 0。配置为数字输入的引脚将根据施密特触发器输入规范转换模拟输入信号。
- 注 2:** 对被定义为数字输入的任何引脚施加模拟电平，均可能会使输入缓冲器的电流消耗超过规定值。

PIC18F45J10 系列

图 18-3: 比较器输出框图



18.6 比较器中断

一旦两个比较器中的任意一个的输出值发生了变化，就会将相应比较器的中断标志位置 1。当从 CMCON<7:6> 读取数据时，需要用软件来保持输出位的状态信息以判断实际发生的变化。CMIF 位（PIR2<6>）位是比较器中断标志位。CMIF 位必须通过清零复位。因为也可以向 CMCON 寄存器写入 1，所以可以模拟中断的发生。

必须将 CMIE 位（PIE2<6>）和 PEIE 位（INTCON<6>）置 1 以允许中断。此外，也必须将 GIE（INTCON<7>）位置 1。只要这些位中的任何一位被清零，虽然当有中断条件产生时 CMIF 位仍会置 1，但却仍然禁止中断。

注： 当执行读操作时（Q2 周期开始时），如果 CMCON 寄存器（C1OUT 或 C2OUT）发生变化，那么 CMIF（PIR2 寄存器）中断标志位就可能不会被置 1。

用户可用以下方式在中断服务程序中清除该中断：

- 对 CMCON 的任何读或写均将中止电平不匹配状态。
- 将标志位 CMIF 清零。

引脚上电平不匹配的情况会一直不断地将 CMIF 标志位置 1。读 CMCON 寄存器将结束引脚上电平不匹配的情况，并允许将 CMIF 标志位清零。

18.7 比较器在休眠模式下的工作方式

当比较器处于活动状态而器件处于休眠模式时，比较器仍可正常工作并可使用比较器中断（如果使能的话）。在允许中断时，中断会把器件从休眠模式唤醒。每个比较器工作时都会消耗额外的电流，如比较器规范中所示。若要把休眠状态下的功耗减少到最小，可在进入休眠模式前关闭比较器模块（CM2:CM0 = 111）。器件从休眠模式唤醒时，CMCON 寄存器的内容不受影响。

18.8 复位的影响

器件复位强制 CMCON 寄存器进入复位状态，导致比较器模块被关闭（CM2:CM0 = 111）。但是，在器件复位时输入引脚（RA0 到 RA3）被默认配置为模拟输入。PCFG3:PCFG0 位（ADCON1<3:0>）的设置决定这些引脚的 I/O 配置。因此，当复位时引脚呈现模拟输入状态，此时器件电流达到最小。

18.9 模拟输入连接注意事项

图 18-4 所示为一个简化的模拟输入电路。由于模拟引脚和数字输出端相连，因而它们与 VDD 和 VSS 之间加有反向偏置的二极管，从而将模拟输入电压限制在 VSS 和 VDD 之间。一旦输入电压超出该范围 0.6V 以上，就

会有一个二极管正偏从而使输入电压被钳位。模拟信号源的最大阻抗值推荐为 10 kΩ。任何连接到模拟输入引脚的外部元件（如电容和齐纳二极管等）的泄漏电流应该极小。

图 18-4: 比较器模拟输入典型电路

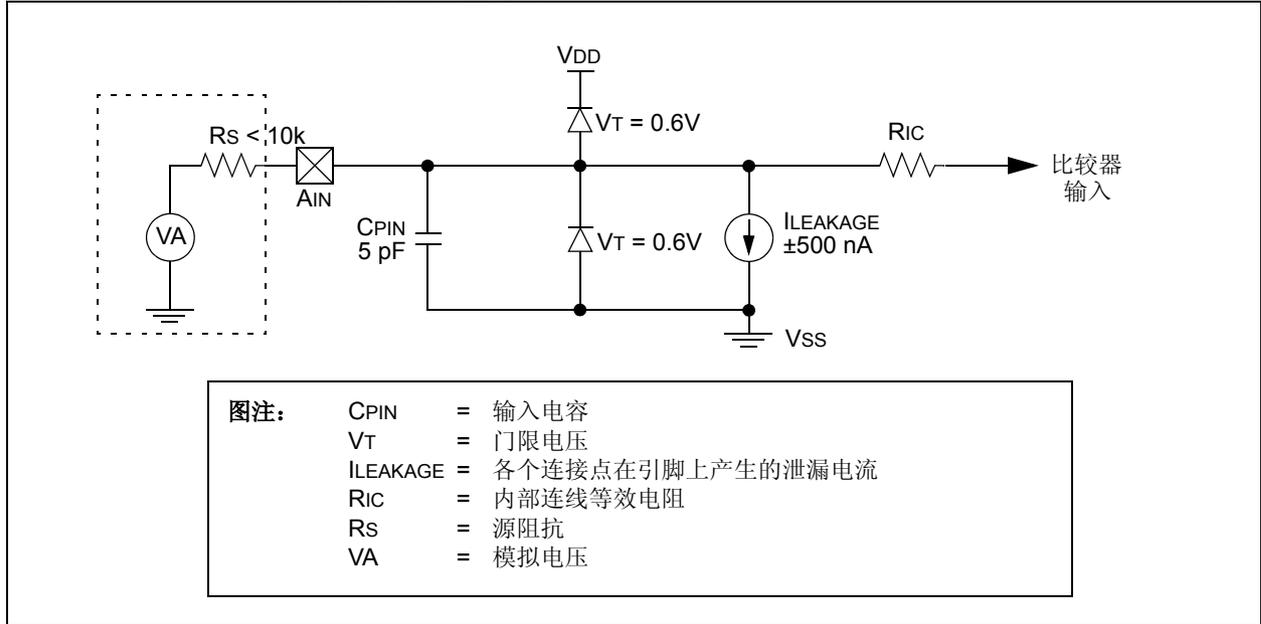


表 18-1: 与比较器模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	45
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	45
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	46
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	—	CCP2IF	45
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	—	CCP2IE	45
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	—	CCP2IP	45
PORTA	—	—	RA5	—	RA3	RA2	RA1	RA0	46
LATA	—	—	PORTA 数据锁存器 (读取和写入数据锁存器)						46
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46

图注: — = 未用, 读为 0。比较器模块不使用阴影单元。

PIC18F45J10 系列

注:

19.0 比较器参考电压模块

比较器参考电压模块是一个 16 阶的梯形电阻网络，提供多个参考电压以供选择。虽然它的主要作用是给模拟比较器提供参考电压，但也可将它用于其他场合。

图 19-1 给出了该模块的框图。梯形电阻分成两组可提供两种量程范围的 CVREF 值，并且还具有限电功能以在不使用参考电压时降低功耗。模块的供电参考电压由器件 VDD/VSS 或外部参考电压提供。

19.1 配置比较器参考电压

参考电压模块由 CVRCON 寄存器（寄存器 19-1）控制，可提供两种范围的输出电压，每种范围都具有 16 种不同的电平。CVRR 位（CVRCON<5>）选择要

用的电压范围。这两种范围的主要区别在于 CVREF 选择位（CVR3:CVR0）选定的步长不同，其中一个范围具有更高的分辨率。下面是计算比较器参考电压输出值的公式：

$$\text{如果 CVRR} = 1: \\ \text{CVREF} = ((\text{CVR3:CVR0})/24) \times \text{CVRSRC}$$

$$\text{如果 CVRR} = 0: \\ \text{CVREF} = (\text{CVRSRC} \times 1/4) + (((\text{CVR3:CVR0})/32) \times \text{CVRSRC})$$

比较器参考模块供电电压可以来自 VDD 和 VSS，或者与 RA2 和 RA3 复用的外部 VREF+ 和 VREF-。CVRSS 位（CVRCON<4>）用于选择电压源。

在改变 CVREF 输出值时，必须考虑比较器参考电压的稳定时间（见第 23.0 节“电气规范”中的表 23-3）。

寄存器 19-1:

CVRCON: 比较器参考电压控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE ⁽¹⁾	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7						bit 0	

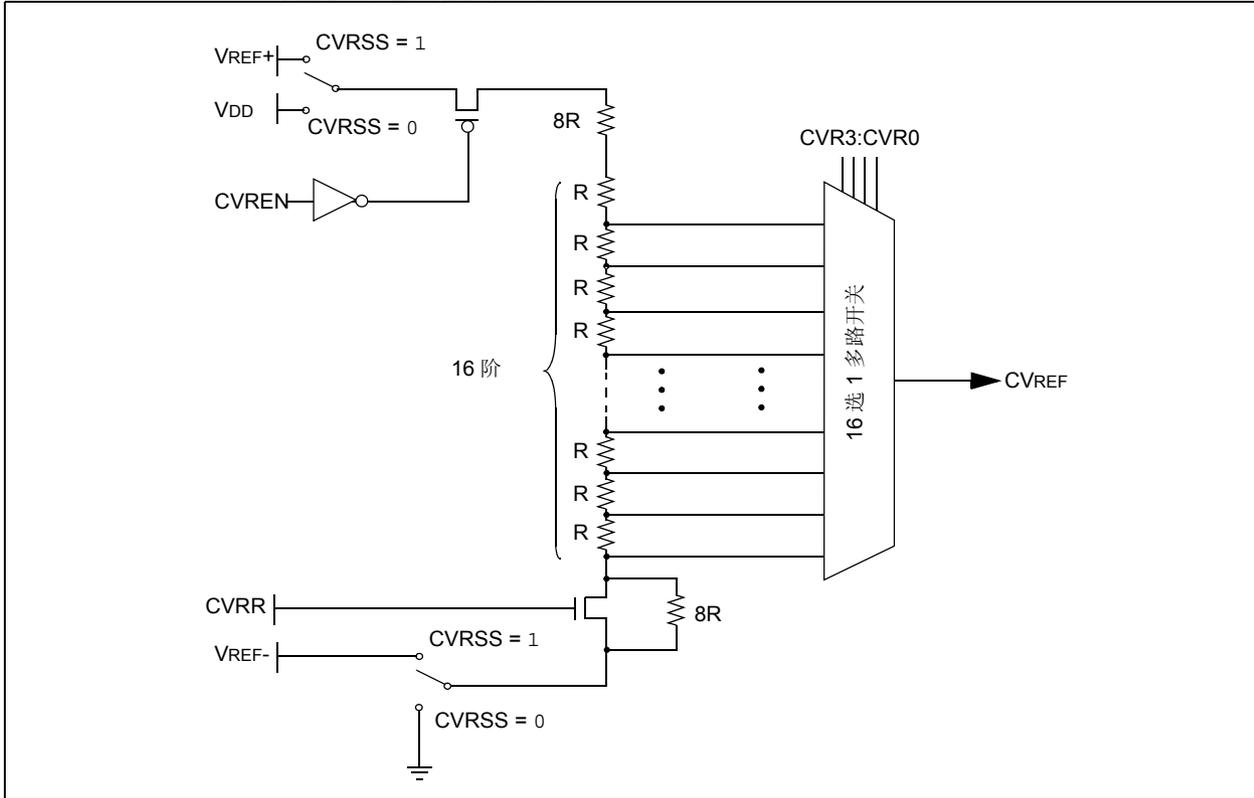
- bit 7 **CVREN:** 比较器参考电压使能位
1 = 开启 CVREF 电路
0 = 关闭 CVREF 电路
- bit 6 **CVROE:** 比较器 VREF 输出使能位 ⁽¹⁾
1 = 在 RA2/AN2/VREF-/CVREF 引脚上输出 CVREF 电压
0 = CVREF 电压与 RA2/AN2/VREF-/CVREF 引脚断开
 注 1: CVROE 的优先级高于 TRISA<2> 位。
- bit 5 **CVRR:** 比较器 VREF 范围选择位
1 = 0 到 0.667 CVRSRC，步长为 CVRSRC/24（低电压范围）
0 = 0.25 CVRSRC 到 0.75 CVRSRC，步长为 CVRSRC/32（高电压范围）
- bit 4 **CVRSS:** 比较器 VREF 源选择位
1 = 比较器参考电压源，CVRSRC = (VREF+) - (VREF-)
0 = 比较器参考电压源，CVRSRC = VDD - VSS
- bit 3-0 **CVR3:CVR0:** 比较器 VREF 值选择位 (0 ≤ (CVR3:CVR0) ≤ 15)
 当 CVRR = 1 时:
 CVREF = ((CVR3:CVR0) / 24) • (CVRSRC)
 当 CVRR = 0 时:
 CVREF = (CVRSRC / 4) + ((CVR3:CVR0) / 32) • (CVRSRC)

图注:

R = 可读位	W = 可写位	U = 未用位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零 x = 未知

PIC18F45J10 系列

图 19-1: 比较器参考电压模块框图



19.2 参考电压精度 / 误差

由于模块结构的原因，模块无法实现满量程参考电压。梯形电阻网络中顶端和底部的晶体管（图 19-1）使 CVREF 无法达到参考电压源的满幅值。参考电压来自于参考电压源；因此，CVREF 输出电平会随参考电压源一起波动。第 23.0 节“电气规范”中可找到经测试得到的参考电压绝对精度值。

19.3 在休眠模式下的工作方式

当中断或看门狗定时器超时唤醒器件时，CVRCON 寄存器内容不受影响。为了最大限度降低休眠模式下的电流消耗，应关闭参考电压模块。

19.4 复位的影响

器件复位通过清零 CVREN（CVRCON<7>）从而禁止参考电压模块；通过清零 CVROE 位（CVRCON<6>），复位还可将参考电压与 RA2 引脚断开，并且通过清零 CVRR（CVRCON<5>），可选择高电压范围。同时 CVR 值选择位也被清零。

19.5 连接注意事项

参考电压模块的工作独立于比较器模块。如果 CVROE 位置 1，参考电压发生器的输出会连到 RA2 引脚。参考电压输出与 RA2 引脚相连时，如果该引脚被配置为数字输入引脚，将会增大电流消耗。使能 CVRSS 时将 RA2 用作数字输入引脚也将增加电流消耗。

RA2 引脚可被直接用作 D/A 输出，但是其驱动能力有限。由于驱动能力有限，当参考电压输出模块连有外部 VREF 时必须使用缓冲器。图 19-2 举例说明了这一缓冲技术。

图 19-2: 比较器参考电压输出缓冲示例

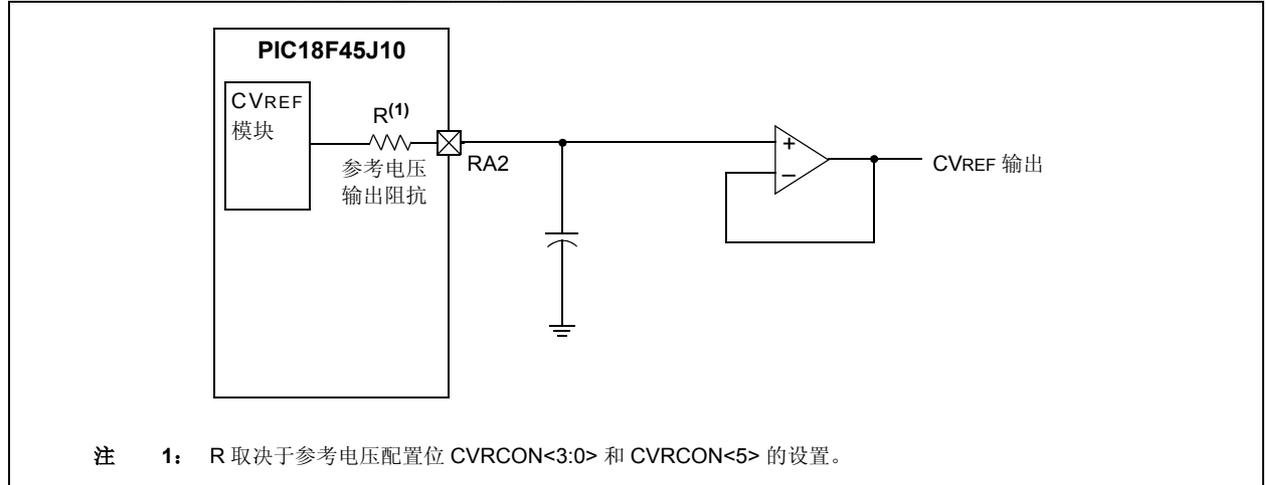


表 19-1: 与比较器参考电压模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	45
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	45
TRISA	—	—	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	46

图注: 比较器参考电压模块不使用阴影单元。

PIC18F45J10 系列

注:

20.0 CPU 的特殊性能

PIC18F45J10 系列器件包含的功能旨在最大限度地提高系统可靠性，并通过减少外部元件将成本降至最低。这些功能包括：

- 振荡器选择
- 复位：
 - 上电复位（POR）
 - 上电延时定时器（PWRT）
 - 振荡器起振定时器（OST）
 - 欠压复位（BOR）
- 中断
- 看门狗定时器（WDT）
- 故障保护时钟监视器
- 双速启动
- 代码保护
- 在线串行编程

根据具体应用对频率、功耗、精度和成本的要求配置振荡器。在第 2.0 节“振荡器配置”中详细讨论了所有的选项。

在本数据手册的前面几章中已经完整地讨论了器件的复位和中断。

PIC18F45J10 系列器件除了为复位提供上电延时定时器和振荡器起振定时器之外，还具有一个可配置的看门狗定时器，该定时器由软件控制。

器件自带的 RC 振荡器还提供了故障保护时钟监视器（FSCM）和双速启动这两个额外的有益功能。FSCM 对外设时钟进行后台监视，并在外设时钟发生故障时自动切换时钟源。双速启动功能使得代码几乎可在起振发生时立即执行，此时主时钟源正在完成其自身的起振延时。

通过设置相应的配置寄存器位可以使能和配置所有这些功能。

20.1 配置位

可以通过对配置位编程（读为 0）或不编程（读为 1）来选择不同的器件配置。这些配置位被映射到程序存储器中从 300000h 开始的单元中。表 20-1 列出了所有的配置位。从寄存器 20-1 到寄存器 20-6 详细解释了各配置位的不同功能。

注意地址 300000h 超出了用户程序存储空间的范围。事实上，它属于配置存储空间（300000h – 3FFFFFFh），这一空间仅能通过表读和表写进行访问。

20.1.1 配置 PIC18F45J10 系列器件的注意事项

与大多数 PIC18 单片机不同，PIC18F45J10 系列器件不使用非易失性存储寄存器存储配置信息。配置字节以易失性存储方式实现，这就意味着在器件每次上电时都必须对配置数据进行编程。

配置数据存储在片内程序存储空间顶部的 4 个字中，这些字被称为闪存配置字。配置位数据按表 20-1 中相同的次序存储在程序存储器中，CONFIG1L 位于地址最低的单元，CONFIG3H 位于地址最高的单元。在器件上电时这些数据被自动装入正确的配置寄存器。

当为这些器件创建应用程序时，用户应该为配置数据特别分配闪存配置字单元，以确保当编译代码时程序代码不会存储在该地址上。

在上电复位时用于配置位的易失性存储单元始终复位为 1。对于其他类型的复位事件，将保留和使用先前已编程的值，而无需从程序存储器重新装入数据。

程序存储器中 CONFIG1H、CONFIG2H 和 CONFIG3H 的高 4 位也应为 1111。这样当这些配置字被远程事件意外执行时，被当作一条 NOP 指令。由于配置位在对应的单元中是未实现的，因此向这些单元写 1 不会影响器件工作。

为了避免在代码执行期间配置被意外更改，可编程配置位只可被写入一次。在上电周期内对其进行初始化之后就不能再次写入了。改变器件的配置需要对器件重新上电。

PIC18F45J10 系列

表 20-1: 配置位和器件 ID

寄存器名称		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	缺省 / 未编程的值 ⁽¹⁾
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	—	—	—	WDTEN	111- ---1
300001h	CONFIG1H	— (2)	— (2)	— (2)	— (2)	— (3)	CP0	—	—	---- x1--
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0	11-- -111
300003h	CONFIG2H	— (2)	— (2)	— (2)	— (2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L	—	—	—	—	—	—	—	—	---- ----
300005h	CONFIG3H	— (2)	— (2)	— (2)	— (2)	—	—	—	CCP2MX	---- ---1
3FFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx ⁽⁴⁾
3FFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0001 110x ⁽⁴⁾

图注: x = 未知, u = 不变, — = 未用。阴影单元未使用, 读为 0。

- 注
- 1: 这些值反映出厂时和上电复位后的未编程状态。在所有其他复位后, 配置字节保持原先的编程状态。
 - 2: 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元, 将会执行 NOP 指令。
 - 3: 该位应始终保持为 0。
 - 4: 请参见寄存器 20-7 和寄存器 20-8 查询 DEVID 的值。这些寄存器为只读寄存器, 用户不能对其进行编程。

寄存器 20-1:

CONFIG1L: 配置寄存器 1 的低字节 (字节地址: 300000h)

R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0	U-0	R/WO-1
DEBUG	XINST	STVREN	—	—	—	—	WDTEN
bit 7							bit 0

- bit 7 **DEBUG:** 后台调试器使能位
 1 = 禁止后台调试器, RB6 和 RB7 被配置为通用 I/O 引脚
 0 = 使能后台调试器, RB6 和 RB7 专用于在线调试
- bit 6 **XINST:** 扩展指令集使能位
 1 = 使能指令集扩展和变址寻址模式
 0 = 禁止指令集扩展和变址寻址模式 (传统模式)
- bit 5 **STVREN:** 堆栈上溢 / 下溢复位使能位
 1 = 使能堆栈上溢 / 下溢复位
 0 = 禁止堆栈上溢 / 下溢复位
- bit 4-1 未用: 读为 0
- bit 0 **WDTEN:** 看门狗定时器使能位
 1 = 使能 WDT
 0 = 禁止 WDT (控制位为 SWDTEN 位)

图注:

R = 可读位 WO = 一次性写入位 U = 未用位, 读为 0
 -n = 未对器件编程时的值 1 = 置 1 0 = 清零

寄存器 20-2:

CONFIG1H: 配置寄存器 1 的高字节 (字节地址: 300001h)

U-0	U-0	U-0	U-0	U-0	R/WO-1	U-0	U-0
—	—	—	—	— (1)	CP0	—	—
bit 7					bit 0		

- bit 7-3 未用: 读为 0
- bit 2 **CP0:** 代码保护位
 1 = 程序存储器未受代码保护
 0 = 程序存储器受代码保护
- bit 1-0 未用: 读为 0
- 注 1: 该位应始终保持为 0。

图注:

R = 可读位 WO = 一次性写入位 U = 未用位, 读为 0
 -n = 未对器件编程时的值 1 = 置 1 0 = 清零

PIC18F45J10 系列

寄存器 20-3: CONFIG2L: 配置寄存器 2 的低字节 (字节地址: 300002h)

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7					bit 0		

- bit 7 **IESO**: 双速启动 (内部 / 外部振荡器切换) 控制位
 1 = 使能双速启动
 0 = 禁止双速启动
- bit 6 **FCMEN**: 故障保护时钟监视器使能位
 1 = 使能故障保护时钟监视器
 0 = 禁止故障保护时钟监视器
- bit 5-3 **未用**: 读为 0
- bit 2 **FOSC2**: 默认 / 复位系统时钟选择位
 1 = 当 OSCCON<1:0> = 00 时, 使用由 FOSC1:FOSC0 选择的时钟作为系统时钟
 0 = 当 OSCCON<1:0> = 00 时使用 INTRC 作为系统时钟
- bit 1-0 **FOSC1:FOSC0**: 振荡器选择位
 11 = EC 振荡器, PLL 使能并由软件控制, OSC2 用作 CLKO 功能
 10 = EC 振荡器, OSC2 用作 CLKO 功能
 01 = HS 振荡器, PLL 使能并由软件控制
 00 = HS 振荡器

图注:

R = 可读位 WO = 一次性写入位 U = 未用位, 读为 0
 -n = 未对器件编程时的值 1 = 置 1 0 = 清零

寄存器 20-4: CONFIG2H: 配置寄存器 2 的高字节 (字节地址: 300003h)

U-0	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7				bit 0			

- bit 7-4 **未用**: 读为 0
- bit 3-0 **WDTPS3:WDTPS0**: 看门狗定时器后分频比选择位
 1111 = 1:32,768
 1110 = 1:16,384
 1101 = 1:8,192
 1100 = 1:4,096
 1011 = 1:2,048
 1010 = 1:1,024
 1001 = 1:512
 1000 = 1:256
 0111 = 1:128
 0110 = 1:64
 0101 = 1:32
 0100 = 1:16
 0011 = 1:8
 0010 = 1:4
 0001 = 1:2
 0000 = 1:1

图注:

R = 可读位 WO = 一次性写入位 U = 未用位, 读为 0
 -n = 未对器件编程时的值 1 = 置 1 0 = 清零

寄存器 20-5: CONFIG3L: 配置寄存器 3 的低字节 (字节地址: 300004h)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
-	-	-	-	-	-	-	-
bit 7							bit 0

bit 7-0 未用: 读为 0

图注:

R = 可读位

U = 未用位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

寄存器 20-6: CONFIG3H: 配置寄存器 3 的高字节 (字节地址: 300005h)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/WO-1
-	-	-	-	-	-	-	CCP2MX
bit 7							bit 0

bit 7-1 未用: 读为 0

bit 0 **CCP2MX:** CCP2 复用位

1 = CCP2 与 RC1 复用

0 = CCP2 与 RB3 复用

图注:

R = 可读位

WO = 一次性写入位

U = 未用位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

PIC18F45J10 系列

寄存器 20-7: PIC18F45J10 系列器件的器件 ID 寄存器 1

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0

bit 7 bit 0

bit 7-5 **DEV2:DEV0:** 器件 ID 位

011 = PIC18F4XJ10

010 = PIC18F2XJ10

001 = PIC18F4XJ10

000 = PIC18F2XJ10

注: DEV10:DEV3 的值可能会与其他器件系列共享使用。使用整个 DEV10:DEV0 位序列标识某一特定器件。

bit 4-0 **REV4:REV0:** 版本 ID 位

这些位用于表明器件版本。

图注:

R = 只读位

U = 未用位, 读为 0

-n = 未对器件编程时的值

u = 编程后状态不变

寄存器 20-8: PIC18F45J10 系列器件的器件 ID 寄存器 2

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3

bit 7 bit 0

bit 7-0 **DEV10:DEV3:** 器件 ID 位

这些位与器件 ID 寄存器 1 中的 DEV2:DEV0 结合使用以标识器件号。

0001 1100 = PIC18FX5J10 器件

0001 1101 = PIC18FX4J10 器件

注: DEV10:DEV3 的值可能会与其他器件系列共享使用。通过使用整个 DEV10:DEV0 位序列标识某一特定器件。

图注:

R = 只读位

U = 未用位, 读为 0

-n = 未对器件编程时的值

u = 编程后状态不变

20.2 看门狗定时器 (WDT)

对于 PIC18F45J10 系列器件, WDT 由 INTRC 振荡器驱动。当使能 WDT 时, 也将使能时钟源。WDT 超时溢出周期的标称值为 4 ms, 其稳定性与 INTRC 振荡器相同。

4 ms 的 WDT 超时溢出周期与 16 位的后分频比值相乘。通过配置寄存器 2H 中的 WDTPS 位控制一个多路开关以对 WDT 后分频器的输出进行选择, 可获得的超时溢出周期范围为 4 ms 至 131.072 秒 (2.18 分钟)。当发生以下任一事件时, WDT 和后分频器将被清零, 这些事件包括: 执行 SLEEP 或 CLRWDT 指令或是发生时钟故障 (主时钟或 Timer1 振荡器)。

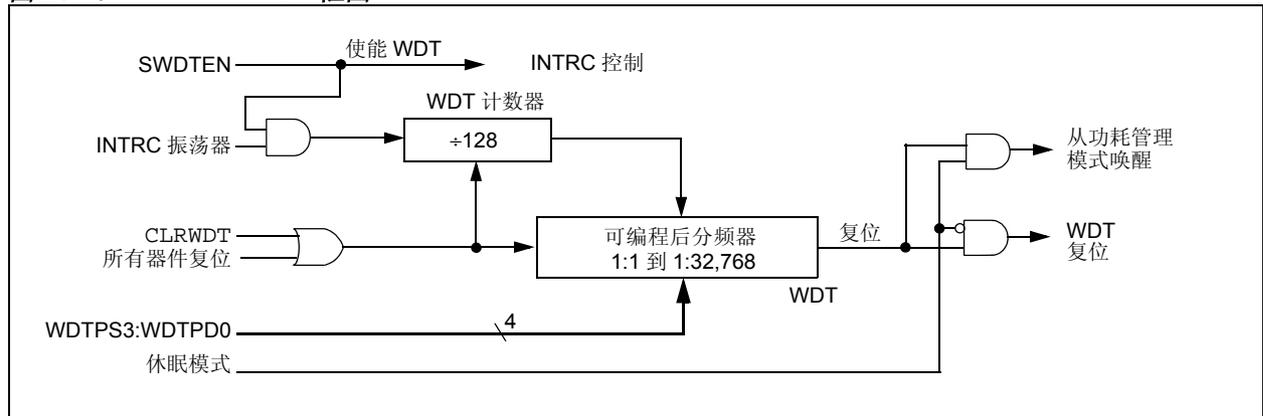
注 1: 当执行 CLRWDT 和 SLEEP 指令时, WDT 和后分频器的计数值将被清零。

注 2: 当执行 CLRWDT 指令时, 后分频器的计数值将被清零。

20.2.1 控制寄存器

WDTCON 寄存器 (寄存器 20-9) 为可读写寄存器。SWDTEN 位使能或禁止 WDT 的操作。

图 20-1: WDT 框图



寄存器 20-9: WDTCON: 看门狗定时器控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN ⁽¹⁾
bit 7							bit 0

bit 7-1 未用: 读为 0

bit 0 **SWDTEN:** 由软件控制的看门狗定时器使能位 ⁽¹⁾

1 = 使能看门狗定时器
0 = 禁止看门狗定时器

注 1: 当 WDTEN 配置位使能时, 该位不起作用。

图注:

R = 可读位 W = 可写位 U = 未用位, 读为 0
-n = 上电复位时的值 1 = 置 1 0 = 清零 x = 未知

表 20-2: 看门狗定时器寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
RCON	IPEN	—	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}	44
WDTCON	—	—	—	—	—	—	—	SWDTEN	44

图注: — = 未用 (读为 0)。看门狗定时器不使用阴影单元。

PIC18F45J10 系列

20.3 片内稳压器

注： 只有标有“F”的器件（如 PIC18F45J10）才有片内稳压器。

标有“LF”的器件内核可由独立于 VDD 的外部电源供电，或者由 VDD 提供输入电压的片内稳压器供电。两个电源共用 VDDCORE/VCAP 引脚。

在“F”器件中，低 ESR 电容必须连接到 VDDCORE/VCAP 引脚，才能使器件正常工作。在标有“LF”编号的器件（即 PIC18LF45J10）中，必须通过 VDDCORE/VCAP 引脚向内核供电。在所有的供电引脚上连接足够的电容始终一种良好的设计习惯。图 20-2 给出了这样的示例。

注： 在标有“LF”的器件中（如 PIC18LF45J10），VDDCORE 不能超过 VDD。

第 23.4 节“AC（时序）规范”列出了内核电压和电容的规范。

20.3.1 片内稳压器和 BOR

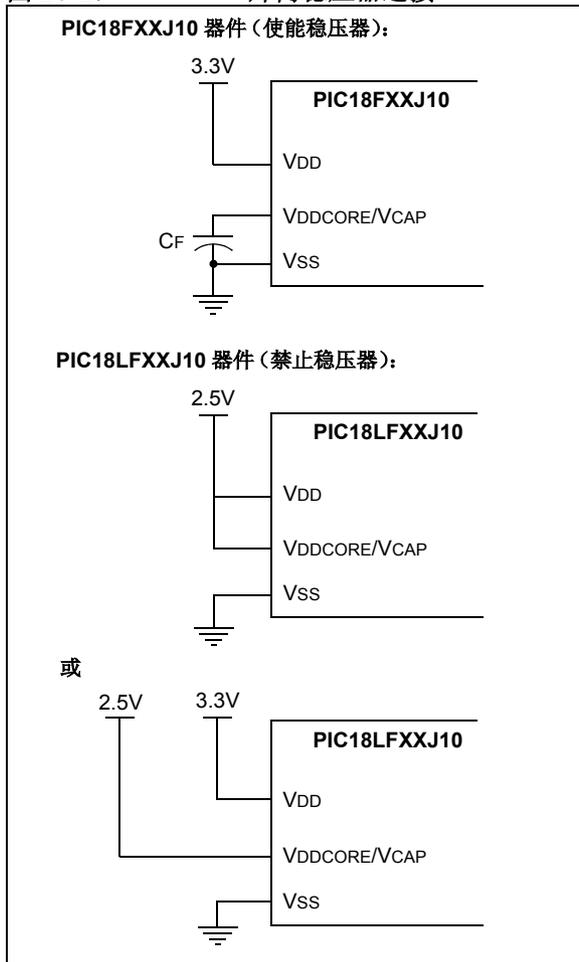
PIC18F45J10 系列器件（标有“F”器件号）也有一个简单的欠压保护功能。如果向稳压器提供的电压不足以维持一个稳定的电平，那么稳压器复位电路将产生 BOR 复位。BOR 标志位（RCON<0>）捕捉该事件。

第 4.4 节“欠压复位（BOR）（仅 PIC18F2XJ10/4XJ10 器件）”和第 4.4.1 节“检测 BOR”详细描述了欠压复位的原理。第 23.1 节“直流规范”指定了欠压电平值。

20.3.2 上电要求

片内稳压器是为了满足器件的上电要求而设计的。在上电时，VDDCORE 决不能比 VDD 高出 0.3V 以上。

图 20-2: 片内稳压器连接



20.4 双速启动

双速启动功能允许单片机在主时钟源可用之前使用 INTRC 振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。通过将 IESO 配置位置 1 可使能该功能。

只有在主振荡器模式是 HS（基于晶振）模式时才可使能双速启动。由于 EC 模式不需 OST 起振延时，因此应禁止双速启动。

当使能双速启动时，在发生上电复位并且上电延时定时器发生超时后，器件复位和从休眠模式唤醒都会使器件将自身配置为使用内部振荡器电路作为时钟源。这样几乎可使代码在主振荡器起振、OST 运行的同时立即执行。一旦 OST 超时，器件就自动切换到 PRI_RUN 模式。

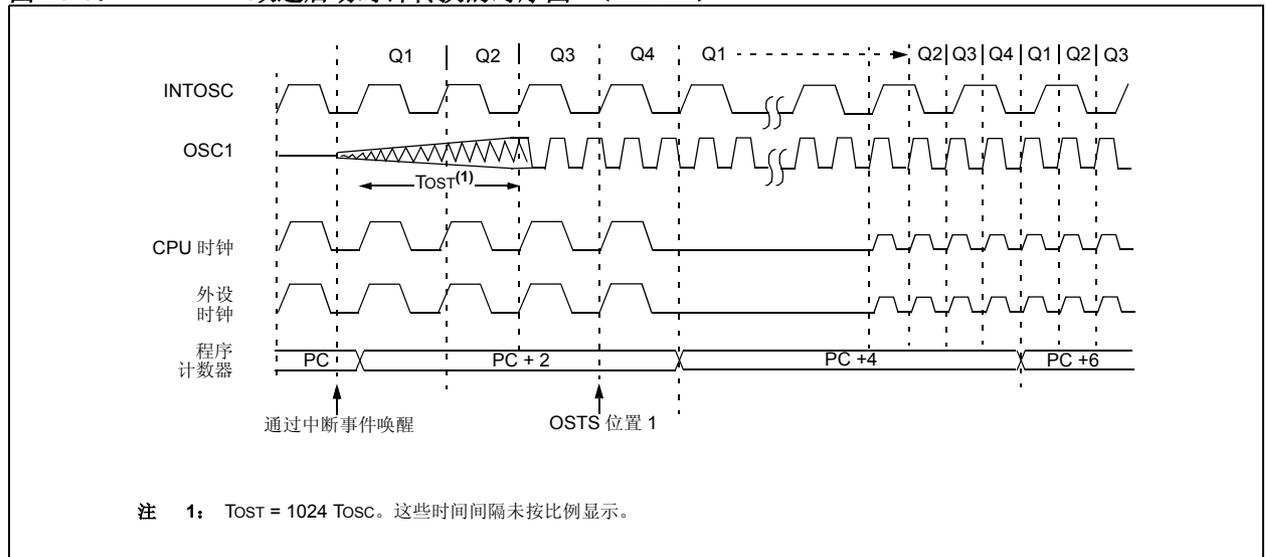
在其他功耗管理模式下不使用双速启动。器件将使用当前选定的时钟源直到主时钟源可用为止。该操作与 IESO 位的设置无关。

20.4.1 使用双速启动时需特别注意的事项

当在双速启动中使用 INTRC 振荡器时，器件仍将遵守进入功耗管理模式的正常命令序列，包括执行多条 SLEEP 指令（见第 3.1.4 节“多条 Sleep 命令”）。实际上，这意味着在 OST 超时前用户代码可以改变 SCS1:SCS0 位的设置或执行 SLEEP 指令。这就使应用程序能短暂地唤醒器件，执行“日常事务”，并在器件开始使用主时钟源前返回休眠状态。

用户代码还能通过检查 OST 位（OSCCON<3>）的状态来确定当前主时钟源是否正在为系统提供时钟。若该位置 1，则表示主振荡器正在为系统提供时钟。否则，表示当器件从复位或休眠模式唤醒期间由内部振荡器电路为系统提供时钟。

图 20-3: 双速启动时钟转换的时序图 (INTRC)



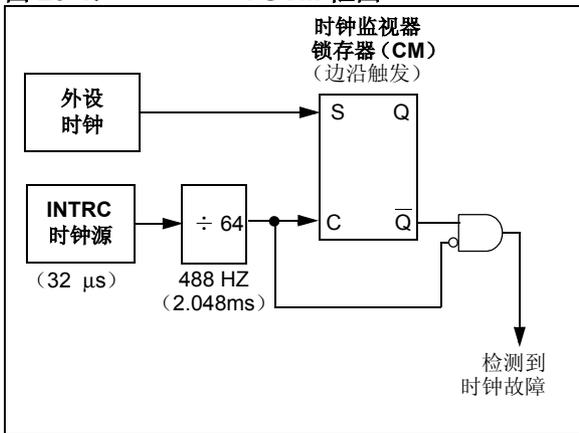
PIC18F45J10 系列

20.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）可使单片机在发生外部时钟故障时，自动将系统时钟切换到内部振荡器电路以保持器件继续运行。将FCMEN配置位置1可使能FSCM功能。

当使能FSCM时，INTRC振荡器将一直保持运行以监视外设时钟，并且在外设时钟发生故障时立即提供备用时钟。时钟监视（如图20-4所示）通过创建一个采样时钟信号实现，该信号为INTRC输出的64分频。这样就使得FSCM采样时钟沿之间有充足的时间间隔，从而保证在此时间段内必然会有外设时钟沿出现。外设器件时钟和采样时钟作为时钟监视器锁存器（CM）的输入。CM在器件时钟的下降沿被置1，在采样时钟的上升沿被清零。

图 20-4: FSCM 框图



在采样时钟的下降沿检测外部时钟故障。如果在出现采样时钟的下降沿时，CM仍置1，就表示检测到外部时钟故障（图20-5）。这将引发以下事件：

- 通过将OSCFIF（PIR2<7>）置1，由FSCM产生振荡器故障中断；
- 器件时钟源切换到内部振荡器电路（OSCCON不会被更新，因此无法显示当前时钟源，这就是故障保护状态）；
- WDT复位。

切换过程中，对于时序要求较高的应用，内部振荡器电路的后分频频率可能不够稳定。在这些情况下，最好选择另一种时钟配置并进入其他功耗管理模式。可以尝试部分恢复或执行安全的关闭。更多详情，请参见第3.1.4节“多条Sleep命令”和第20.4.1节“使用双速启动时需特别注意的事项”。

为了在唤醒器件时使用更快的时钟速率，可以选择INTOSC或后分频器时钟源以提供更快的时钟速率，这可以通过在复位发生后立即设置IRCF2:IRCF0实现。对于从休眠模式唤醒的情况，可以通过在进入休眠模式之前设置IRCF2:IRCF0来选择INTOSC或后分频器时钟源。

FSCM只能检测出主时钟源或辅助时钟源的故障。如果内部振荡器电路发生故障，将无法被检测到，当然也就不可能采取任何措施。

20.5.1 FSCM 和看门狗定时器

FSCM和WDT均以INTRC振荡器作为时钟源。由于WDT使用独立的分频器和计数器，当使能FSCM时，禁止WDT不影响INTRC振荡器的运行。

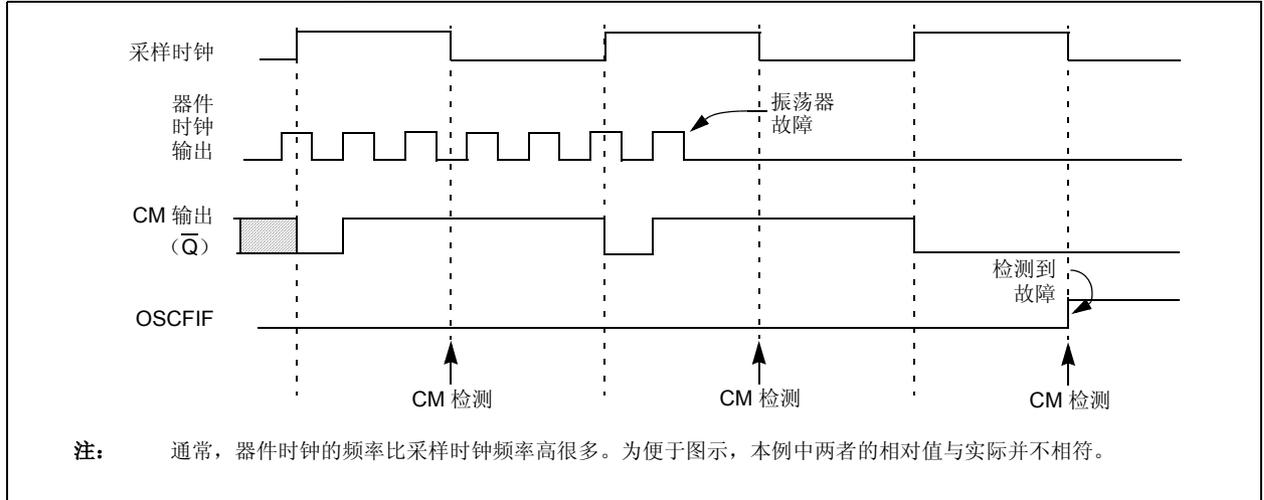
如前所述，当检测到时钟故障时，时钟将切换到INTRC时钟源；这可能意味着代码执行速度会发生很大变化。如果用小分频值使能WDT，时钟速率的下降将使WDT发生超时并在随后使器件复位。基于这个原因，故障保护事件也会使WDT和后分频器复位，从而使WDT从执行速度发生变化的那一刻开始重新计数，因而减少了发生错误超时的可能性。

20.5.2 退出故障保护运行模式

器件复位或进入功耗管理模式均可终止故障保护状态。发生复位时，控制器启动配置寄存器2H中指定的主时钟源（如果运行在HS模式下，将伴有OST振荡器起振延时）。INTRC振荡器将在主时钟源就绪前提供系统时钟（类似于双速启动）。当主时钟源可用时，系统时钟源将切换回主时钟（OSCCON寄存器中的OSTS位置1以表明当前使用的是主时钟源）。然后，故障保护时钟监视器恢复对外设时钟的监视。

在启动时，主时钟源可能永远不能就绪。在这种情况下，器件运行将以INTRC振荡器作为时钟源。OSCCON寄存器将保持复位状态直到进入功耗管理模式为止。

图 20-5: FSCM 时序图



20.5.3 功耗管理模式下的 FSCM 中断

进入功耗管理模式时，时钟多路开关选择由 `OSCCON` 寄存器选定的时钟源。在功耗管理模式下将恢复对功耗管理时钟源的故障保护监视。

如果在功耗管理运行期间发生了振荡器故障，随后将会发生的事件取决于是否允许了振荡器故障中断。如果允许 (`OSCFIF = 1`)，代码执行将以 `INTOSC` 复用器作为时钟源，并不会自动转回到发生故障的时钟源。

如果禁止该中断，空闲模式下故障所导致的中断将使 CPU 开始执行指令，此时由 `INTOSC` 时钟源提供时钟。

20.5.4 上电复位或从休眠模式唤醒

FSCM 在器件退出上电复位 (POR) 或低功耗休眠模式后的任一时刻都可以检测到振荡器故障。当系统主时钟为 `EC` 或 `INTRC` 模式时，监视会在这些事件发生后立即开始。

对于 `HS` 模式，情况会有所不同。由于这类振荡器需要的起振时间可能比 FSCM 采样时钟的时间长很多，因此可能会检测到假的时钟故障。为了避免这一情况，内部振荡器电路会被自动配置为器件时钟并一直工作直到主时钟稳定下来为止 (`OST` 定时器已完成延时)。这与双速启动模式相同。一旦主时钟稳定下来，`INTRC` 就将重新作为 FSCM 时钟源。

注：防止在发生上电复位或从休眠状态唤醒时发生假中断的电路同样也将阻止在发生这些事件后对振荡器故障的检测。通过监视 `OSTS` 位，并使用定时程序来确定振荡器起振时间是否过长可避免这个问题。即使如此，在检测到振荡器故障时也不会再在振荡器故障中断标志位上有所反映。

正如第 20.4.1 节“使用双速启动时需特别注意的事项”中所述，在等待主时钟稳定的过程中，可以选择另一种时钟配置并进入某一功耗管理模式。当选择新的功耗管理模式时，主时钟将被禁止。

PIC18F45J10 系列

20.6 程序校验和代码保护

对于 PIC18F45J10 系列中的所有器件，片内程序存储器空间视为一个独立的存储区。配置位 CP0 控制该存储区的代码保护。该位阻止对程序存储空间的外部读写，但对正常的执行模式没有直接影响。

20.6.1 配置寄存器保护

有两种方法保护配置寄存器使其免遭破坏性的改写或读取。主要的保护方式是配置位的一次性写入功能，该功能阻止对在上电周期内完成编程的位再次进行配置。要阻止不可预见的事件，由于电池故障（如 ESD 事件）产生的配置位更改将导致奇偶校验错误并触发器件复位。

配置寄存器的数据来自于程序存储器中的闪存配置字。当 CP0 位置 1 时，也将保护器件配置的源数据。

20.7 在线串行编程

PIC18F45J10 系列单片机可以在最终应用电路中进行串行编程。只需要 5 根线即可完成这一操作，其中时钟线和数据线各一根，其余 3 根分别是电源线、接地线和编程电压线。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程，从而使固件版本保持最新或定制固件。

20.8 在线调试器

将 DEBUG 配置位清零可启用在线调试功能。这一功能允许使用 MPLAB® IDE 进行一些简单的调试。当启用了单片机的这项功能时，某些资源就不再是通用的了。表 20-3 显示了后台调试器所需的资源。

表 20-3: 调试器资源

I/O 引脚:	RB6 和 RB7
堆栈:	2 级
程序存储器:	512 字节
数据存储器:	32 字节

21.0 指令集综述

PIC18F45J10 系列器件具有一个含有 75 条 PIC18 内核指令的标准指令集，和一个含有优化递归代码或利用软件堆栈的 8 条新指令的扩展指令集。本章后面的部分将讨论该扩展指令集。

21.1 标准指令集

标准的 PIC18 指令集与以前的 PICmicro® 指令集相比，添加了很多增强功能，并保持了易于从其他 PICmicro 指令集移植的特点。大部分指令为单字指令（16 位），只有 4 条指令是双字指令。

每条单字指令都是一个 16 位字，包括指定指令类型的操作码和指定指令具体操作的一个或多个操作数。

整个指令集具有高度的正交性，分为以下 4 种基本类型：

- 面向字节的操作类指令
- 面向位的操作类指令
- 立即数操作类指令
- 控制操作类指令

表 21-1 中的 PIC18 指令集汇总列出了面向字节、面向位、立即数和控制操作类指令。表 21-1 给出了对操作码字段的说明。

大部分面向字节的操作类指令含有三种操作数：

1. 文件寄存器（由“f”指定）
2. 保存结果的目标寄存器（由“d”指定）
3. 被访问的存储器（由“a”指定）

文件寄存器指示符“f”指定了指令将会使用哪一个文件寄存器。目标寄存器指示符“d”指定了操作结果的存放位置。如果“d”为 0，操作结果存入 WREG 寄存器中；如果“d”为 1，操作结果存入指令指定的文件寄存器中。

所有面向位的操作类指令都含有三种操作数：

1. 文件寄存器（由“f”指定）
2. 文件寄存器中的位（由“b”指定）
3. 被访问的存储器（由“a”指定）

位域指示符“b”指定操作所影响的位的编号，而文件寄存器指示符“f”则代表这些位所在的寄存器的编号。

立即数操作指令使用以下操作数：

- 要装入文件寄存器中的立即数（由“k”指定）
- 要装入立即数的 FSR 寄存器（由“f”指定）
- 不需要操作数（由“—”指定）

控制类指令使用以下操作数：

- 程序存储器地址（由“n”指定）
- CALL 或 RETURN 指令的模式（由“s”指定）
- 表读和表写指令的模式（由“m”指定）
- 不需要操作数（由“—”指定）

除了 4 条双字指令外，所有的指令都是单字指令。双字指令将所需的信息保存在 32 位中。第二个字的高 4 位都是 1。如果第二个字作为一条指令执行，它会执行 NOP 指令。

除非条件测试结果为真或者指令执行改变了程序计数器的值，否则执行所有的单字指令都只需要一个指令周期。对于上述两种特殊情况，执行指令需要两个指令周期，第二个指令周期中执行一条 NOP 指令。

执行双字指令需要两个指令周期。

每个指令周期由 4 个振荡器周期组成。因此，对于频率为 4 MHz 的振荡器，其正常的指令执行时间为 1 μs。如果条件测试结果为真或指令执行改变了程序计数器的值，则该指令的执行时间为 2 μs。双字跳转指令（如果条件测试结果为真）的执行需要 3 μs。

图 21-1 给出了指令的几种通用格式。所有示例均使用“nnh”来表示十六进制数。

指令集汇总（见表 21-2）列出了可被 Microchip 汇编器（MPASM™）识别的标准指令。

第 21.1.1 节“标准指令集”中对每条指令进行了说明。

PIC18F45J10 系列

表 21-1: 操作码字段说明

字段	说明
a	快速操作 RAM 位 a = 0: 快速操作 RAM 内的 RAM 单元 (BSR 寄存器被忽略) a = 1: RAM 存储区由 BSR 寄存器指定
bbb	某 8 位文件寄存器内的位地址 (0 到 7)。
BSR	存储区选择寄存器。用于选择当前的 RAM 存储区。
C, DC, Z, OV 和 N	ALU 状态位: 进位标志位 、 辅助进位标志位 、 全零标志位 、 溢出标志位 和 负标志位
d	目标寄存器选择位 d = 0: 结果保存至 WREG 寄存器 d = 1: 结果保存至文件寄存器 f
dest	目标单元: 可以是 WREG 寄存器或指定的文件寄存器单元。
f	8 位寄存器地址 (00h 到 FFh) 或 2 位 FSR 指示符 (0h 到 3h)。
f _s	12 位寄存器地址 (000h 到 FFFh)。这是源地址。
f _d	12 位寄存器地址 (000h 到 FFFh)。这是目标地址。
GIE	全局中断允许位。
k	立即数、常数或者标号 (可能是 8 位、12 位或 20 位的值)。
label	标号名称。
mm	用于表读和表写指令的 TBLPTR 寄存器的模式。 仅供读表和写表指令使用:
*	不改变寄存器 (如用于表读和表写的 TBLPTR)。
*+	后增寄存器 (如用于表读和表写的 TBLPTR)
*-	后减寄存器 (如用于表读和表写的 TBLPTR)
++	预增寄存器 (如用于表读和表写的 TBLPTR)
n	相对跳转指令的相对地址 (二进制补码) 或 Call/Branch 和 Return 指令的直接地址。
PC	程序计数器。
PCL	程序计数器的低字节。
PCH	程序计数器的高字节。
PCLATH	程序计数器的高字节锁存器。
PCLATU	程序计数器的最高字节锁存器。
PD	掉电位。
PRODH	乘积的高字节。
PRODL	乘积的低字节。
s	快速调用 / 返回模式选择位。 s = 0: 不对影子寄存器进行更新, 也不用影子寄存器的内容更新其他寄存器 s = 1: 将某些寄存器的值存入影子寄存器或把影子寄存器的值载入某些寄存器 (快速模式)
TBLPTR	21 位表指针 (指向程序存储器单元)。
TABLAT	8 位表锁存器。
TU	超时溢出位。
TOS	栈顶。
u	未使用或不变。
WDT	看门狗定时器。
WREG	工作寄存器 (累加器)。
x	任意值 (0 或 1)。编译器将产生 x = 0 的代码。为了与所有的 Microchip 软件工具兼容, 建议使用这种格式。
z _s	对寄存器 (源) 进行间接寻址的 7 位偏移量。
z _d	对寄存器 (目标) 进行间接寻址的 7 位偏移量。
{ }	可选参数。
[text]	表示变址地址。
(text)	text 的内容。
[expr]<n>	表示由指针 expr 指定的寄存器中的位 n。
→	赋值。
< >	寄存器位域。
∈	表示属于某个集合。
斜体文字	用户定义项 (字体为 Courier)。

图 21-1: 指令的通用格式

<p>面向字节的文件寄存器操作</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">10</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">9</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td style="text-align: center;">d</td> <td style="text-align: center;">a</td> <td colspan="7" style="text-align: center;">f(寄存器地址)</td> </tr> </table> <p>d = 0 表示结果存入 WREG 寄存器 d = 1 表示结果存入文件寄存器 (f) a = 0 强制使用快速操作存储区 a = 1 根据 BSR 选择存储区 f = 8 位文件寄存器地址</p>		15		10		9		8		7		0	操作码		d	a	f(寄存器地址)							<p>指令示例</p> <p>ADDWF MYREG, W, B</p>																		
15		10		9		8		7		0																																
操作码		d	a	f(寄存器地址)																																						
<p>字节到字节的传送操作 (双字)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="8" style="text-align: center;">f(源寄存器地址)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">1111</td> <td colspan="8" style="text-align: center;">f(目标寄存器地址)</td> </tr> </table> <p>f = 12 位文件寄存器地址</p>		15		12		11					0	操作码		f(源寄存器地址)								15		12		11					0	1111		f(目标寄存器地址)								<p>MOVFF MYREG1, MYREG2</p>
15		12		11					0																																	
操作码		f(源寄存器地址)																																								
15		12		11					0																																	
1111		f(目标寄存器地址)																																								
<p>面向位的文件寄存器操作</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">9</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="3" style="text-align: center;">b(位地址)</td> <td style="text-align: center;">a</td> <td colspan="6" style="text-align: center;">f(寄存器地址)</td> </tr> </table> <p>b = 占 3 位, 表示文件寄存器 (f) 中位的位置 a = 0 强制使用快速操作存储区 a = 1 根据 BSR 选择存储区 f = 8 位文件寄存器地址</p>		15		12		11		9		8		7		0	操作码		b(位地址)			a	f(寄存器地址)						<p>BSF MYREG, bit, B</p>															
15		12		11		9		8		7		0																														
操作码		b(位地址)			a	f(寄存器地址)																																				
<p>立即数操作</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="7" style="text-align: center;">k(立即数)</td> </tr> </table> <p>k = 8 位立即数的值</p>		15				8		7			0	操作码		k(立即数)							<p>MOVLW 7Fh</p>																					
15				8		7			0																																	
操作码		k(立即数)																																								
<p>控制操作</p> <p>CALL、GOTO 和跳转类操作</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="8" style="text-align: center;">n<7:0>(立即数)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">1111</td> <td colspan="8" style="text-align: center;">n<19:8>(立即数)</td> </tr> </table> <p>n = 20 位立即数的值</p>		15				8		7			0	操作码		n<7:0>(立即数)								15		12		11					0	1111		n<19:8>(立即数)								<p>GOTO Label</p>
15				8		7			0																																	
操作码		n<7:0>(立即数)																																								
15		12		11					0																																	
1111		n<19:8>(立即数)																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td style="text-align: center;">S</td> <td colspan="7" style="text-align: center;">n<7:0>(立即数)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">1111</td> <td colspan="8" style="text-align: center;">n<19:8>(立即数)</td> </tr> </table> <p>S = 快速位</p>		15				8		7			0	操作码		S	n<7:0>(立即数)							15		12		11					0	1111		n<19:8>(立即数)								<p>CALL MYFUNC</p>
15				8		7			0																																	
操作码		S	n<7:0>(立即数)																																							
15		12		11					0																																	
1111		n<19:8>(立即数)																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">10</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="8" style="text-align: center;">n<10:0>(立即数)</td> </tr> </table>		15				11		10			0	操作码		n<10:0>(立即数)								<p>BRA MYFUNC</p>																				
15				11		10			0																																	
操作码		n<10:0>(立即数)																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">操作码</td> <td colspan="8" style="text-align: center;">n<7:0>(立即数)</td> </tr> </table>		15				8		7			0	操作码		n<7:0>(立即数)								<p>BC MYFUNC</p>																				
15				8		7			0																																	
操作码		n<7:0>(立即数)																																								

PIC18F45J10 系列

表 21-2: PIC18FXXXX 指令集

助记符, 操作数	说明	指令周期数	16 位指令字				受影响的状态位	注	
			MSb			LSb			
面向字节的操作类指令									
ADDWF	f, d, a	WREG 与 f 相加	1	0010	01da	ffff	ffff	C、DC、Z、OV、N	1, 2
ADDWFC	f, d, a	WREG 与 f 带进位相加	1	0010	00da	ffff	ffff	C、DC、Z、OV、N	1, 2
ANDWF	f, d, a	WREG 与 f 作逻辑与运算	1	0001	01da	ffff	ffff	Z、N	1, 2
CLRF	f, a	将 f 清零	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	将 f 取反	1	0001	11da	ffff	ffff	Z、N	1, 2
CPFSEQ	f, a	将 f 与 WREG 作比较, 相等则跳过	1 (2 或 3)	0110	001a	ffff	ffff	无	4
CPFSGT	f, a	将 f 与 WREG 作比较, 大于则跳过	1 (2 或 3)	0110	010a	ffff	ffff	无	4
CPFSLT	f, a	将 f 与 WREG 作比较, 小于则跳过	1 (2 或 3)	0110	000a	ffff	ffff	无	1, 2
DECF	f, d, a	f 减 1	1	0000	01da	ffff	ffff	C、DC、Z、OV、N	1, 2, 3, 4
DECFSZ	f, d, a	f 减 1, 为 0 则跳过	1 (2 或 3)	0010	11da	ffff	ffff	无	1, 2, 3, 4
DCFSNZ	f, d, a	f 减 1, 非 0 则跳过	1 (2 或 3)	0100	11da	ffff	ffff	无	1, 2
INCF	f, d, a	f 加 1	1	0010	10da	ffff	ffff	C、DC、Z、OV、N	1, 2, 3, 4
INCFSZ	f, d, a	f 加 1, 为 0 则跳过	1 (2 或 3)	0011	11da	ffff	ffff	无	4
INFSNZ	f, d, a	f 加 1, 非 0 则跳过	1 (2 或 3)	0100	10da	ffff	ffff	无	1, 2
IORWF	f, d, a	WREG 与 f 作逻辑或运算	1	0001	00da	ffff	ffff	Z、N	1, 2
MOVF	f, d, a	传送 f	1	0101	00da	ffff	ffff	Z、N	1
MOVFF	f _s , f _d	将 f _s (源) 地址放入第 1 个字 将 f _d (目标) 地址放入第 2 个字	2	1100	ffff	ffff	ffff	无	
MOVWF	f, a	将 WREG 传送到 f	1	0110	111a	ffff	ffff	无	
MULWF	f, a	WREG 乘以 f	1	0000	001a	ffff	ffff	无	1, 2
NEGF	f, a	将 f 取补	1	0110	110a	ffff	ffff	C、DC、Z、OV、N	
RLCF	f, d, a	对 f 执行带进位循环左移	1	0011	01da	ffff	ffff	C、Z、N	1, 2
RLNCF	f, d, a	f 循环左移 (不带进位)	1	0100	01da	ffff	ffff	Z、N	
RRCF	f, d, a	对 f 执行带进位循环右移	1	0011	00da	ffff	ffff	C、Z、N	
RRNCF	f, d, a	f 循环右移 (不带进位)	1	0100	00da	ffff	ffff	Z、N	
SETF	f, a	将 f 置为全 1	1	0110	100a	ffff	ffff	无	1, 2
SUBFWB	f, d, a	WREG 减去 f (带借位)	1	0101	01da	ffff	ffff	C、DC、Z、OV、N	
SUBWF	f, d, a	f 减去 WREG	1	0101	11da	ffff	ffff	C、DC、Z、OV、N	1, 2
SUBWFB	f, d, a	f 减去 WREG (带借位)	1	0101	10da	ffff	ffff	C、DC、Z、OV、N	
SWAPF	f, d, a	将 f 中的两个半字节进行交换	1	0011	10da	ffff	ffff	无	4
TSTFSZ	f, a	测试 f, 为 0 则跳过	1 (2 或 3)	0110	011a	ffff	ffff	无	1, 2
XORWF	f, d, a	WREG 与 f 作逻辑异或运算	1	0001	10da	ffff	ffff	Z、N	

- 注 1: 端口寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 它使用引脚上的当前值更新自身内容。例如, 如果将一引脚配置为输入, 其对应数据锁存器中的值为 1, 但此时若有外部器件将该引脚拉为低电平, 则被写回数据锁存器的数据值将是 0。
- 2: 当该指令针对 TMR0 寄存器执行时 (指令中的 d = 1), 如果已为其分配了预分频器, 则将该分频器清零。
- 3: 如果执行一条程序计数器 (PC) 被修改或者条件检测为真的指令, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- 4: 某些指令是双字指令。如果指令的第一个字无法取得后面 16 位中嵌入的信息, 则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元内存储的都是合法的指令。

PIC18F45J10 系列

表 21-2: PIC18FXXXX 指令集 (续)

助记符, 操作数	说明	指令周期数	16 位指令字				受影响的状态位	注	
			MSb			LSb			
面向位的操作类指令									
BCF	f, b, a	将 f 寄存器中的某位清零	1	1001	bbba	ffff	ffff	无	1, 2
BSF	f, b, a	将 f 寄存器中的某位置 1	1	1000	bbba	ffff	ffff	无	1, 2
BTFSC	f, b, a	测试 f 中的某位, 为 0 则跳过	1 (2 或 3)	1011	bbba	ffff	ffff	无	3, 4
BTFSS	f, b, a	测试 f 中的某位, 为 1 则跳过	1 (2 或 3)	1010	bbba	ffff	ffff	无	3, 4
BTG	f, d, a	将 f 中的位取反	1	0111	bbba	ffff	ffff	无	1, 2
控制操作类指令									
BC	n	进位则跳转	1 (2)	1110	0010	nnnn	nnnn	无	4
BN	n	为负则跳转	1 (2)	1110	0110	nnnn	nnnn	无	
BNC	n	无进位则跳转	1 (2)	1110	0011	nnnn	nnnn	无	
BNN	n	不为负则跳转	1 (2)	1110	0111	nnnn	nnnn	无	
BNOV	n	不溢出则跳转	1 (2)	1110	0101	nnnn	nnnn	无	
BNZ	n	不为零则跳转	1 (2)	1110	0001	nnnn	nnnn	无	
BOV	n	溢出则跳转	1 (2)	1110	0100	nnnn	nnnn	无	
BRA	n	无条件跳转	2	1101	0nnn	nnnn	nnnn	无	
BZ	n	为零则跳转	1 (2)	1110	0000	nnnn	nnnn	无	
CALL	n, s	调用子程序 (第一个字)	2	1110	110s	kkkk	kkkk	无	
		(第二个字)		1111	kkkk	kkkk	kkkk	无	
CLRWDT	—	将看门狗定时器清零	1	0000	0000	0000	0100	TO, PD	
DAW	—	对 WREG 进行十进制调整	1	0000	0000	0000	0111	C	
GOTO	n	跳转到地址 (第一个字)	2	1110	1111	kkkk	kkkk	无	
		(第二个字)		1111	kkkk	kkkk	kkkk	无	
NOP	—	空操作	1	0000	0000	0000	0000	无	
NOP	—	空操作	1	1111	xxxx	xxxx	xxxx	无	
POP	—	弹出返回堆栈栈顶 (TOS) 的内容	1	0000	0000	0000	0110	无	
PUSH	—	压入返回堆栈栈顶 (TOS)	1	0000	0000	0000	0101	无	
RCALL	n	相对调用	2	1101	1nnn	nnnn	nnnn	无	
RESET	—	用软件使器件复位	1	0000	0000	1111	1111	全部	
RETFIE	s	中断返回使能	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	返回时将立即数送入 WREG	2	0000	1100	kkkk	kkkk	无	
RETURN	s	从子程序返回	2	0000	0000	0001	001s	无	
SLEEP	—	进入待机模式	1	0000	0000	0000	0011	TO, PD	

- 注 1: 端口寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 它使用引脚上的当前值更新自身内容。例如, 如果将一引脚配置为输入, 其对应数据寄存器中的值为 1, 但此时若有外部器件将该引脚拉为低电平, 则被写回数据寄存器的数据值将是 0。
- 2: 当该指令针对 TMR0 寄存器执行时 (指令中的 d = 1), 如果已为其分配了预分频器, 则将该分频器清零。
- 3: 如果执行一条程序计数器 (PC) 被修改或者条件检测为真的指令, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- 4: 某些指令是双字指令。如果指令的第一个字无法取得后面 16 位中嵌入的信息, 则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元内存储的都是合法的指令。

PIC18F45J10 系列

表 21-2: PIC18FXXXX 指令集(续)

助记符, 操作数	说明	指令周期数	16 位指令字				受影响的 状态位	注	
			MSb			LSb			
立即数操作类指令									
ADDLW	k	WREG 与立即数相加	1	0000	1111	kkkk	kkkk	C、DC、Z、OV、N	
ANDLW	k	立即数与 WREG 作逻辑与运算	1	0000	1011	kkkk	kkkk	Z、N	
IORLW	k	立即数与 WREG 作逻辑或运算	1	0000	1001	kkkk	kkkk	Z、N	
LFSR	f, k	将立即数 (12 位) (第二个字) 传送到 FSR(f) (第一个字)	2	1110	1110	00ff	kkkk	无	
MOVLB	k	将立即数传送到 BSR<3:0>	1	0000	0001	0000	kkkk	无	
MOVLW	k	将立即数传送到 WREG	1	0000	1110	kkkk	kkkk	无	
MULLW	k	立即数与 WREG 相乘	1	0000	1101	kkkk	kkkk	无	
RETLW	k	返回时将立即数送入 WREG	2	0000	1100	kkkk	kkkk	无	
SUBLW	k	立即数减去 WREG	1	0000	1000	kkkk	kkkk	C、DC、Z、OV、N	
XORLW	k	立即数与 WREG 作逻辑异或运算	1	0000	1010	kkkk	kkkk	Z、N	
数据存储器 ↔ 程序存储器操作类指令									
TBLRD*		表读	2	0000	0000	0000	1000	无	
TBLRD*+		后增表读		0000	0000	0000	1001	无	
TBLRD*-		后减表读		0000	0000	0000	1010	无	
TBLRD+*		预增表读		0000	0000	0000	1011	无	
TBLWT*		表写	2	0000	0000	0000	1100	无	
TBLWT*+		后增表写		0000	0000	0000	1101	无	
TBLWT*-		后减表写		0000	0000	0000	1110	无	
TBLWT+*		预增表写		0000	0000	0000	1111	无	

- 注 1: 端口寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 它使用引脚上的当前值更新自身内容。例如, 如果将一引脚配置为输入, 其对应数据锁存器中的值为 1, 但此时若有外部器件将该引脚拉为低电平, 则被写回数据锁存器的数据值将是 0。
- 2: 当该指令针对 TMR0 寄存器执行时 (指令中的 d = 1), 如果已为其分配了预分频器, 则将该分频器清零。
- 3: 如果执行一条程序计数器 (PC) 被修改或者条件检测为真的指令, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- 4: 某些指令是双字指令。如果指令的第一个字无法取得后面 16 位中嵌入的信息, 则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元内存储的都是合法的指令。

21.1.1 标准指令集

ADDLW 将 w 的内容与立即数相加

语法: ADDLW k

操作数: $0 \leq k \leq 255$

操作: $(W) + k \rightarrow W$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0000	1111	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与 8 位立即数 “k” 相加, 结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

示例: ADDLW 15h

执行指令前
W = 10h
执行指令后
W = 25h

ADDWF 将 w 与 f 相加

语法: ADDWF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W) + (f) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0010	01da	ffff	ffff
------	------	------	------

说明: 将 W 的内容与 f 寄存器的内容相加。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: ADDWF REG, 0, 0

执行指令前
W = 17h
REG = 0C2h
执行指令后
W = 0D9h
REG = 0C2h

注: 所有的 PIC18 指令都可能在其指令助记符之前使用可选的标号参数, 用于在符号寻址中使用。如果使用了标号, 那么指令语法将变为: {label} 指令参数。

PIC18F45J10 系列

ADDWFC

w 与 f 带进位相加

语法: ADDWFC f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W) + (f) + (C) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0010	00da	ffff	ffff
------	------	------	------

说明: 将 W 的内容、进位标志位与数据存储器单元 f 相加。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存储在寄存器 f 中。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: ADDWFC REG, 0, 1

执行指令前

进位标志位 = 1
 REG = 02h
 W = 4Dh

执行指令后

进位标志位 = 0
 REG = 02h
 W = 50h

ANDLW

将立即数和 W 寄存器作逻辑与运算

语法: ANDLW k

操作数: $0 \leq k \leq 255$

操作: $(W) .AND. k \rightarrow W$

受影响的状态位: N 和 Z

机器码:

0000	1011	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与 8 位立即数 k 作逻辑与运算。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例: ANDLW 05Fh

执行指令前

W = A3h

执行指令后

W = 03h

PIC18F45J10 系列

BCF 将 f 寄存器中的某位清零

语法: BCF f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

操作: $0 \rightarrow f \langle b \rangle$

受影响的状态位: 无

机器码:

1001	bbba	ffff	ffff
------	------	------	------

说明: 将寄存器 f 中的位 b 清零。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: BCF FLAG_REG, 7, 0

执行指令前
 FLAG_REG = C7h
 执行指令后
 FLAG_REG = 47h

BN 为负则跳转

语法: BN n

操作数: $-128 \leq n \leq 127$

操作: 如果负标志位为 1
 (PC) $+2+2n \rightarrow PC$

受影响的状态位: 无

机器码:

1110	0110	nnnn	nnnn
------	------	------	------

说明: 如果负标志位为 1, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 $PC + 2 + 2n$ 。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:
 如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BN Jump

执行指令前
 PC = 地址 (HERE)
 执行指令后
 如果负标志位 = 1;
 PC = 地址 (Jump)
 如果负标志位 = 0;
 PC = 地址 (HERE + 2)

BNC 无进位则跳转

语法: BNC n

操作数: $-128 \leq n \leq 127$

操作: 如果进位标志位为 0
(PC) +2+2n → PC

受影响的状态位: 无

机器码:

1110	0011	nnnn	nnnn
------	------	------	------

说明: 如果进位标志位为 0, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BNC Jump

执行指令前
PC = 地址 (HERE)

执行指令后
如果进位标志位 = 0;
PC = 地址 (Jump)
如果进位标志位 = 1;
PC = 地址 (HERE + 2)

BNN 不为负则跳转

语法: BNN n

操作数: $-128 \leq n \leq 127$

操作: 如果负标志位为 0
(PC) +2+2n → PC

受影响的状态位: 无

机器码:

1110	0111	nnnn	nnnn
------	------	------	------

说明: 如果负标志位为 0, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BNN Jump

执行指令前
PC = 地址 (HERE)

执行指令后
如果负标志位 = 0;
PC = 地址 (Jump)
如果负标志位 = 1;
PC = 地址 (HERE + 2)

PIC18F45J10 系列

BNOV 不溢出则跳转

语法: BNOV n

操作数: $-128 \leq n \leq 127$

操作: 如果溢出标志位为 0
(PC) +2+2n → PC

受影响的状态位: 无

机器码:

1110	0101	nnnn	nnnn
------	------	------	------

说明: 如果溢出标志位为 0, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BNOV Jump

执行指令前

PC = 地址 (HERE)

执行指令后

如果溢出标志位 = 0;
PC = 地址 (Jump)
如果溢出标志位 = 1;
PC = 地址 (HERE + 2)

BNZ 不为零则跳转

语法: BNZ n

操作数: $-128 \leq n \leq 127$

操作: 如果全零标志位为 0
(PC) +2+2n → PC

受影响的状态位: 无

机器码:

1110	0001	nnnn	nnnn
------	------	------	------

说明: 如果全零标志位为 0, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BNZ Jump

执行指令前

PC = 地址 (HERE)

执行指令后

如果全零标志位 = 0;
PC = 地址 (Jump)
如果全零标志位 = 1;
PC = 地址 (HERE + 2)

PIC18F45J10 系列

BTFSC 测试寄存器中的位，为 0 则跳过

语法: BTFSC f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

操作: 如果 $(f < b) = 0$ 则跳过

受影响的状态位: 无

机器码:

1011	bbba	ffff	ffff
------	------	------	------

说明: 如果寄存器 f 中的位 b 为 0, 则跳过下一条指令。即在位 b 为 0 时, 丢弃在当前指令执行期间所取的下一条指令, 而是执行一条 NOP 指令取而代之, 使该指令变成双周期指令。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。
 详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)
注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```
HERE    BTFSC    FLAG, 1, 0
FALSE   :
TRUE    :
```

执行指令前
 PC = 地址 (HERE)

执行指令后
 如果 $FLAG < 1 > = 0$;
 PC = 地址 (TRUE)
 如果 $FLAG < 1 > = 1$;
 PC = 地址 (FALSE)

BTFSS 测试寄存器中的位，为 1 则跳过

语法: BTFSS f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$

操作: 如果 $(f < b) = 1$ 则跳过

受影响的状态位: 无

机器码:

1010	bbba	ffff	ffff
------	------	------	------

说明: 如果寄存器 f 中的位 b 为 1, 则跳过下一条指令。即在位 b 为 1 时, 丢弃在当前指令执行期间所取的下一条指令, 而是执行一条 NOP 指令取而代之, 使该指令变成双周期指令。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。
 详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)
注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```
HERE    BTFSS    FLAG, 1, 0
FALSE   :
TRUE    :
```

执行指令前
 PC = 地址 (HERE)

执行指令后
 如果 $FLAG < 1 > = 0$;
 PC = 地址 (FALSE)
 如果 $FLAG < 1 > = 1$;
 PC = 地址 (TRUE)

BTG 将 f 中的某位取反

语法: BTG f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$

操作: $(\overline{f}) \rightarrow f$

受影响的状态位: 无

机器码:

0111	bbba	ffff	ffff
------	------	------	------

说明: 将数据存储单元 f 中的位 b 取反。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: BTG PORTC, 4, 0

执行指令前:
 PORTC = 0111 0101 [75h]
 执行指令后:
 PORTC = 0110 0101 [65h]

BOV 溢出则跳转

语法: BOV n

操作数: $-128 \leq n \leq 127$

操作: 如果溢出标志位为 1
 $(PC) + 2 + 2n \rightarrow PC$

受影响的状态位: 无

机器码:

1110	0100	nnnn	nnnn
------	------	------	------

说明: 如果溢出标志位为 1, 那么程序将跳转。二进制补码“2n”与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 $PC + 2 + 2n$ 。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BOV Jump

执行指令前
 PC = 地址 (HERE)
 执行指令后
 如果溢出标志位 = 1;
 PC = 地址 (Jump)
 如果溢出标志位 = 0;
 PC = 地址 (HERE + 2)

PIC18F45J10 系列

BZ 为零则跳转

语法: BZ n

操作数: $-128 \leq n \leq 127$

操作: 如果全零标志位为 1
(PC) + 2 + 2n → PC

受影响的状态位: 无

机器码:

1110	0000	nnnn	nnnn
------	------	------	------

说明: 如果全零标志位为 1, 那么程序将跳转。二进制补码 “2n” 与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BZ Jump

执行指令前
PC = 地址 (HERE)

执行指令后
如果全零标志位 = 1;
PC = 地址 (Jump)
如果全零标志位 = 0;
PC = 地址 (HERE + 2)

CALL 调用子程序

语法: CALL k {,s}

操作数: $0 \leq k \leq 1048575$
s ∈ [0,1]

操作: (PC) + 4 → TOS,
k → PC<20:1>,
如果 s = 1
(W) → WS,
(STATUS) → STATUSS,
(BSR) → BSRS

受影响的状态位: 无

机器码:

1110	110s	k ₇ kkk	kkkk ₀
1111	k ₁₉ kkk	kkkk	kkkk ₈

第一个字 (k<7:0>)
第二个字 (k<19:8>)

说明: 可在整个 2 MB 的存储器范围内进行子程序调用。首先, 将返回地址 (PC+4) 压入返回堆栈。如果 s = 1, 则 W、STATUS 和 BSR 寄存器也存入它们各自的影子寄存器 WS、STATUSS 和 BSRS。如果 s = 0, 将不会进行任何更新 (默认)。然后将 “k” 的 20 位值装入 PC<20:1>。CALL 是一条双周期指令。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	将 PC 压入堆栈	读立即数 k<19:8>, 写入 PC
空操作	空操作	空操作	空操作

示例: HERE CALL THERE, 1

执行指令前
PC = 地址 (HERE)

执行指令后
PC = 地址 (THERE)
TOS = 地址 (HERE + 4)
WS = W
BSRS = BSR
STATUSS = STATUS

CLRF 将 f 清零

语法: CLRF f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: $000h \rightarrow f$
 $1 \rightarrow Z$

受影响的状态位: Z

机器码:

0110	101a	ffff	ffff
------	------	------	------

说明: 清零指定寄存器的内容。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写 寄存器 f

示例: CLRF FLAG_REG, 1

执行指令前
 FLAG_REG = 5Ah
 执行指令后
 FLAG_REG = 00h

CLRWDT 将看门狗定时器清零

语法: CLRWDT

操作数: 无

操作: $000h \rightarrow WDT$,
 $000h \rightarrow WDT$ 后分频器,
 $1 \rightarrow \overline{TO}$,
 $1 \rightarrow \overline{PD}$

受影响的状态位: \overline{TO} 和 \overline{PD}

机器码:

0000	0000	0000	0100
------	------	------	------

说明: CLRWDT 指令复位看门狗定时器。而且还会复位 WDT 的后分频器。状态位 \overline{TO} 和 \overline{PD} 被置 1。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	空操作

示例: CLRWDT

执行指令前
 WDT 计数器 = ?
 执行指令后
 WDT 计数器 = 00h
 WDT 后分频器 = 0
 \overline{TO} = 1
 \overline{PD} = 1

PIC18F45J10 系列

COMF

将 f 取反

语法: COMF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(\bar{f}) \rightarrow \text{dest}$

受影响的状态位: N 和 Z

机器码:

0001	11da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容取反。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: COMF REG, 0, 0

执行指令前
REG = 13h
 执行指令后
REG = 13h
 W = ECh

CPFSEQ

比较 f 和 w, 如果 f = w 则跳过

语法: CPFSEQ f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: (f) - (W),
 如果 (f) = (W) 则跳过
 (无符号比较)

受影响的状态位: 无

机器码:

0110	001a	ffff	ffff
------	------	------	------

说明: 通过执行无符号减法, 将数据存储单元 f 的内容与 W 的内容做比较。如果 f = W, 则所取的指令被丢弃, 转而执行一条 NOP 指令, 从而使该指令变成双周期指令。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)

注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例: HERE CPFSEQ REG, 0
 NEQUAL :
 EQUAL :

执行指令前
PC 地址 = HERE
 W = ?
 REG = ?
 执行指令后
 如果 REG = W ;
 PC = 地址 (EQUAL)
 如果 REG ≠ W ;
 PC = 地址 (NEQUAL)

CPFSGT 比较 f 和 W, 如果 f > W 则跳过

语法: CPFSGT f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: (f) - (W),
 如果 (f) > (W) 则跳过
 (无符号比较)

受影响的状态位: 无

机器码:

0110	010a	ffff	ffff
------	------	------	------

说明: 通过执行无符号减法, 将数据存储器单元 f 的内容与 W 的内容做比较。如果 f 的内容大于 WREG 的内容, 则所取的指令会被丢弃, 转而执行一条 NOP 指令, 从而使该指令变成双周期指令。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)

注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```
HERE    CPFSGT REG, 0
NGREATER :
GREATER :
```

执行指令前
 PC = 地址 (HERE)
 W = ?

执行指令后
 如果 REG > W;
 PC = 地址 (GREATER)
 如果 REG ≤ W;
 PC = 地址 (NGREATER)

CPFSLT 比较 f 和 W, 如果 f < W 则跳过

语法: CPFSLT f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: (f) - (W),
 如果 (f) < (W) 则跳过
 (无符号比较)

受影响的状态位: 无

机器码:

0110	000a	ffff	ffff
------	------	------	------

说明: 通过执行无符号减法, 将数据存储器单元 f 的内容与 W 的内容做比较。如果 f 的内容小于 W 的内容, 则所取的指令会被丢弃, 转而执行一条 NOP 指令, 从而使该指令变成双周期指令。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

指令字数: 1

指令周期数: 1(2)

注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```
HERE    CPFSLT REG, 1
NLESS  :
LESS   :
```

执行指令前
 PC = 地址 (HERE)
 W = ?

执行指令后
 如果 REG < W;
 PC = 地址 (LESS)
 如果 REG ≥ W;
 PC = 地址 (NLESS)

PIC18F45J10 系列

DAW	对 W 寄存器进行十进制调整								
语法:	DAW								
操作数:	无								
操作:	如果 $[W<3:0> > 9]$ 或 $[DC = 1]$, 则 $(W<3:0>) + 6 \rightarrow W<3:0>$; 否则 $(W<3:0>) \rightarrow W<3:0>$ 如果 $[W<7:4> + DC > 9]$ 或 $[C = 1]$, 则 $(W<7:4>) + 6 + DC \rightarrow W<7:4>$; 否则 $W<7:4> + DC \rightarrow W<7:4>$								
受影响的状态位:	C								
机器码:	<table border="1"> <tr> <td>0000</td> <td>0000</td> <td>0000</td> <td>0111</td> </tr> </table>	0000	0000	0000	0111				
0000	0000	0000	0111						
说明:	指令 DAW 调整 W 内的 8 位值, 即前一条加法指令中两个变量 (均为压缩的 BCD 格式) 的和, 并产生一个正确的压缩 BCD 格式的结果。								
指令字数:	1								
指令周期数:	1								
Q 周期操作:									
	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>译码</td> <td>读 寄存器 W</td> <td>处理数据</td> <td>写 寄存器 W</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	译码	读 寄存器 W	处理数据	写 寄存器 W
Q1	Q2	Q3	Q4						
译码	读 寄存器 W	处理数据	写 寄存器 W						

例 1: DAW

执行指令前

W = A5h
C = 0
DC = 0

执行指令后

W = 05h
C = 1
DC = 0

例 2:

执行指令前

W = CEh
C = 0
DC = 0

执行指令后

W = 34h
C = 1
DC = 0

DECF	f 减 1								
语法:	DECF f {,d {,a}}								
操作数:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
操作:	$(f) - 1 \rightarrow \text{dest}$								
受影响的状态位:	C、DC、N、OV 和 Z								
机器码:	<table border="1"> <tr> <td>0000</td> <td>01da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0000	01da	ffff	ffff				
0000	01da	ffff	ffff						
说明:	将寄存器 f 的内容减 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。								
指令字数:	1								
指令周期数:	1								
Q 周期操作:									
	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>译码</td> <td>读 寄存器 f</td> <td>处理数据</td> <td>写入 目标寄存器</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	译码	读 寄存器 f	处理数据	写入 目标寄存器
Q1	Q2	Q3	Q4						
译码	读 寄存器 f	处理数据	写入 目标寄存器						

示例: DECF CNT, 1, 0

执行指令前

CNT = 01h
Z = 0

执行指令后

CNT = 00h
Z = 1

PIC18F45J10 系列

GOTO 无条件跳转

语法: GOTO k
 操作数: $0 \leq k \leq 1048575$
 操作: $k \rightarrow PC<20:1>$
 受影响的状态位: 无

机器码:

第一个字 (k<7:0>)	1110	1111	k ₇ kkk	kkkk ₀
第二个字 (k<19:8>)	1111	k ₁₉ kkk	kkkk	kkkk ₈

说明: GOTO 指令允许无条件跳转到整个 2 MB 存储器范围中的任何位置。将 20 位值 k 装入 PC<20:1>。GOTO 始终为一条双周期指令。

指令字数: 2
 指令周期数: 2
 Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	空操作	读立即数 k<19:8>, 写入 PC
空操作	空操作	空操作	空操作

示例: GOTO THERE
 执行指令后
 PC = 地址 (THERE)

INCF f 加 1

语法: INCF f{,d{,a}}
 操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$
 操作: $(f) + 1 \rightarrow \text{dest}$

受影响的状态位: C、DC、N、OV 和 Z

机器码:

0010	10da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容加 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1
 指令周期数: 1
 Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: INCF CNT, 1, 0

执行指令前

CNT	=	FFh
Z	=	0
C	=	?
DC	=	?

执行指令后

CNT	=	00h
Z	=	1
C	=	1
DC	=	1

INCFSZ **f 加 1, 为 0 则跳过**

语法: INCFSZ f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) + 1 \rightarrow \text{dest}$
 如果结果为 0 则跳过

受影响的状态位: 无

机器码:

0011	11da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容加 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
 如果结果为 0, 则丢弃所取的指令, 转而执行一条 NOP 指令, 从而使该指令成为双周期指令。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)
注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```

HERE    INCFSZ    CNT, 1, 0
NZERO   :
ZERO    :
```

执行指令前
 PC = 地址 (HERE)

执行指令后
 CNT = CNT + 1
 如果 CNT = 0;
 PC = 地址 (ZERO)
 如果 CNT ≠ 0;
 PC = 地址 (NZERO)

INFSNZ **f 加 1, 非 0 则跳过**

语法: INFSNZ f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) + 1 \rightarrow \text{dest}$
 如果结果 ≠ 0 则跳过

受影响的状态位: 无

机器码:

0100	10da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容加 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
 如果结果不为 0, 则丢弃所取的指令, 转而执行一条 NOP 指令, 从而使该指令成为双周期指令。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)
注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```

HERE    INFSNZ    REG, 1, 0
ZERO    :
NZERO   :
```

执行指令前
 PC = 地址 (HERE)

执行指令后
 REG = REG + 1
 如果 REG ≠ 0;
 PC = 地址 (NZERO)
 如果 REG = 0;
 PC = 地址 (ZERO)

PIC18F45J10 系列

IORLW 立即数与 w 作逻辑或运算

语法: IORLW k

操作数: $0 \leq k \leq 255$

操作: $(W) .OR. k \rightarrow W$

受影响的状态位: N 和 Z

机器码:

0000	1001	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与 8 位立即数 k 作逻辑或运算。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 寄存器 W

示例: IORLW 35h

执行指令前
W = 9Ah

执行指令后
W = BFh

IORWF w 与 f 作逻辑或运算

语法: IORWF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W) .OR. (f) \rightarrow dest$

受影响的状态位: N 和 Z

机器码:

0001	00da	ffff	ffff
------	------	------	------

说明: 将 W 与寄存器 f 的内容作逻辑或运算。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: IORWF RESULT, 0, 1

执行指令前
RESULT = 13h
W = 91h

执行指令后
RESULT = 13h
W = 93h

LFSR 载入 FSR

语法: LFSR f, k

操作数: $0 \leq f \leq 2$
 $0 \leq k \leq 4095$

操作: $k \rightarrow \text{FSRf}$

受影响的状态位: 无

机器码:

1110	1110	00ff	$k_{11}k_{kk}$
1111	0000	k_7k_{kk}	k_{kkk}

说明: 将 12 位立即数 k 载入由 f 指向的文件选择寄存器。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k 的 MSB	处理数据	写立即数 k 的 MSB 到 FSRfH
译码	读立即数 k 的 LSB	处理数据	将立即数 k 写入 FSRfL

示例: LFSR 2, 3ABh

执行指令后
FSR2H = 03h
FSR2L = ABh

MOVF 传送 f

语法: MOVF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $f \rightarrow \text{dest}$

受影响的状态位: N 和 Z

机器码:

0101	00da	ffff	ffff
------	------	------	------

说明: 根据 d 的状态, 将寄存器 f 的内容传送到目标寄存器。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。f 可以为 256 字节存储区中的任何单元。
如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 W

示例: MOVF REG, 0, 0

执行指令前
REG = 22h
W = FFh
执行指令后
REG = 22h
W = 22h

PIC18F45J10 系列

MOVFF 将源寄存器的内容传至目标寄存器

语法: MOVFF f_s, f_d

操作数: $0 \leq f_s \leq 4095$
 $0 \leq f_d \leq 4095$

操作: $(f_s) \rightarrow f_d$

受影响的状态位: 无

机器码:

第一个字 (源)	1100	ffff	ffff	ffff f_s
第二个字 (目标)	1111	ffff	ffff	ffff f_d

说明: 将源寄存器 f_s 的内容传送到目标寄存器 f_d 。
 源寄存器 f_s 可以是 4096 字节数据空间 (000h 到 FFFh) 中的任何地址单元, 目标寄存器 f_d 也可以是 000h 到 FFFh 中的任何地址单元。
 源或目标寄存器都可以是 W (这是个有用的特例)。
 MOVFF 指令特别适用于将数据存储单元中的内容移入外设寄存器 (如发送缓冲器或 I/O 端口)。
 MOVFF 指令中不能使用 PCL、TOSU、TOSH 或 TOSL 作为目标寄存器。

指令字数: 2

指令周期数: 2 (3)

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f (源)	处理数据	空操作
译码	空操作 非虚拟读取	空操作	写寄存器 f (目标)

示例: MOVFF REG1, REG2

执行指令前
 REG1 = 33h
 REG2 = 11h

执行指令后
 REG1 = 33h
 REG2 = 33h

MOVLB 将立即数传送到 BSR 的低半字节

语法: MOVLW k

操作数: $0 \leq k \leq 255$

操作: $k \rightarrow \text{BSR}$

受影响的状态位: 无

机器码:

0000	0001	kkkk	kkkk
------	------	------	------

说明: 将 8 位立即数 k 装入存储区选择寄存器 (BSR)。不管 $k_7:k_4$ 的值如何, $\text{BSR}\langle 7:4 \rangle$ 的值将始终保持为 0。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	将立即数 k 写入 BSR

示例: MOVLB 5

执行指令前
 BSR 寄存器 = 02h

执行指令后
 BSR 寄存器 = 05h

MOVLW 将立即数传送到 w

语法: `MOVLW k`
 操作数: $0 \leq k \leq 255$
 操作: $k \rightarrow W$
 受影响的状态位: 无
 机器码:

0000	1110	kkkk	kkkk
------	------	------	------

 说明: 将 8 位立即数 k 装入 W。
 指令字数: 1
 指令周期数: 1
 Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

示例: `MOVLW 5Ah`

执行指令后
`W = 5Ah`

MOVWF 将 w 的内容传送到 f

语法: `MOVWF f{,a}`
 操作数: $0 \leq f \leq 255$
 $a \in [0,1]$
 操作: $(W) \rightarrow f$
 受影响的状态位: 无
 机器码:

0110	111a	ffff	ffff
------	------	------	------

 说明: 将 W 寄存器中的数据传送到寄存器 f。
 “f”可以是 256 字节存储区中的任何单元。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1
 指令周期数: 1
 Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写 寄存器 f

示例: `MOVWF REG, 0`

执行指令前
`W = 4Fh`
`REG = FFh`
 执行指令后
`W = 4Fh`
`REG = 4Fh`

PIC18F45J10 系列

MULLW 将立即数和 w 中的内容相乘

语法: MULLW k

操作数: $0 \leq k \leq 255$

操作: $(W) \times k \rightarrow \text{PRODH:PRODL}$

受影响的状态位: 无

机器码:

0000	1101	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与 8 位立即数 k 进行无符号的乘法运算。16 位的结果存储在 PRODH:PRODL 寄存器对中, 其中 PRODH 存储高字节。W 的内容不变。所有状态标志位都不受影响。请注意此操作不可能发生溢出或进位。结果有可能为零, 但不会在相应的标志位上反映出来。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写 寄存器 PRODH: PRODL

示例: MULLW 0C4h

执行指令前

W = E2h
PRODH = ?
PRODL = ?

执行指令后

W = E2h
PRODH = ADh
PRODL = 08h

MULWF 将 w 与 f 相乘

语法: MULWF f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: $(W) \times (f) \rightarrow \text{PRODH:PRODL}$

受影响的状态位: 无

机器码:

0000	001a	ffff	ffff
------	------	------	------

说明: 将 W 的内容与寄存器单元 f 的内容执行无符号的乘法运算。16 位的结果存储在 PRODH:PRODL 寄存器对中, 其中 PRODH 存储高字节。W 和 f 的内容都不变。所有状态标志位都不受影响。请注意此操作不可能发生溢出或进位。结果有可能为零, 但不会在相应的标志位上反映出来。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写 寄存器 PRODH: PRODL

示例: MULWF REG, 1

执行指令前

W = C4h
REG = B5h
PRODH = ?
PRODL = ?

执行指令后

W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

NEGF **对 f 取补**

语法: **NEGF f {,a}**

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: $(\bar{f}) + 1 \rightarrow f$

受影响的状态位: **N、OV、C、DC 和 Z**

机器码:

0110	110a	ffff	ffff
------	------	------	------

说明: 用二进制补码对单元 **f** 取补。结果存储在数据存储单元 **f** 中。
 如果 **a** 为 0，选择快速操作存储区。如果 **a** 为 1，使用 **BSR** 选择 **GPR** 存储区（默认）。
 如果 **a** 为 0 且使能了扩展的指令集，只要 $f \leq 95$ (**5Fh**)，指令就将以立即数变址寻址模式进行操作。详情请参见 **第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数: **1**

指令周期数: **1**

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写 寄存器 f

示例: **NEGF REG, 1**

执行指令前
REG = 0011 1010 [**3Ah**]

执行指令后
REG = 1100 0110 [**C6h**]

NOP **空操作**

语法: **NOP**

操作数: 无

操作: 空操作

受影响的状态位: 无

机器码:

0000	0000	0000	0000
1111	xxxx	xxxx	xxxx

说明: 不执行任何操作。

指令字数: **1**

指令周期数: **1**

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作

示例:
 无。

PIC18F45J10 系列

POP 弹出返回堆栈栈顶的内容

语法:	POP				
操作数:	无				
操作:	(TOS) → 丢弃				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table;"> <tr> <td>0000</td> <td>0000</td> <td>0000</td> <td>0110</td> </tr> </table>	0000	0000	0000	0110
0000	0000	0000	0110		

说明: 从返回堆栈弹出 TOS 值并丢弃。然后, 前一个压入返回堆栈的值成为 TOS 值。此指令可以让用户正确管理返回堆栈, 从而实现软件堆栈。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	弹出 TOS 值	空操作

示例: POP
GOTO NEW

执行指令前
TOS = 0031A2h
堆栈 (向下 1 级) = 014332h

执行指令后
TOS = 014332h
PC = NEW

PUSH 将数据压入返回堆栈栈顶

语法:	PUSH				
操作数:	无				
操作:	(PC+2) → TOS				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table;"> <tr> <td>0000</td> <td>0000</td> <td>0000</td> <td>0101</td> </tr> </table>	0000	0000	0000	0101
0000	0000	0000	0101		

说明: PC+2 的值被压入返回堆栈的栈顶。原先的 TOS 值被压入堆栈的下一级。此指令允许通过修改 TOS 并将其压入返回堆栈来实现软件堆栈。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	将 PC + 2 压入返回堆栈	空操作	空操作

示例: PUSH

执行指令前
TOS = 345Ah
PC = 0124h

执行指令后
PC = 0126h
TOS = 0126h
堆栈 (向下一级) = 345Ah

RCALL 相对调用

语法: RCALL n

操作数: $-1024 \leq n \leq 1023$

操作: (PC) + 2 → TOS,
(PC) + 2 + 2n → PC

受影响的状态位: 无

机器码:

1101	1nnn	nnnn	nnnn
------	------	------	------

说明: 从当前地址跳转 (最多 1 KB) 来调用子程序。首先, 将返回地址 (PC + 2) 压入返回堆栈。然后, 将二进制补码 “2n” 与 PC 相加。因为 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 n 将 PC 压入 堆栈	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例: HERE RCALL Jump

执行指令前
PC = 地址 (HERE)

执行指令后
PC = 地址 (Jump)
TOS = 地址 (HERE + 2)

RESET 复位

语法: RESET

操作数: 无

操作: 将所有受 MCLR 复位影响的寄存器和标志位复位。

受影响的状态位: 全部

机器码:

0000	0000	1111	1111
------	------	------	------

说明: 此指令可实现用软件清零 MCLR 复位。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	开始 复位	空操作	空操作

示例: RESET

执行指令后
寄存器 = 复位值
标志位 * = 复位值

PIC18F45J10 系列

RETFIE 从中断返回

语法: RETFIE {s}

操作数: $s \in [0,1]$

操作: (TOS) → PC,
1 → GIE/GIEH 或 PEIE/GIEL,
如果 $s = 1$
(WS) → W;
(STATUS) → STATUS;
(BSRS) → BSR;
PCLATU 和 PCLATH 保持不变。

受影响的状态位: GIE/GIEH 和 PEIE/GIEL。

0000	0000	0001	000s
------	------	------	------

说明: 从中断返回。执行出栈操作, 将栈顶 (TOS) 的内容装入 PC。通过将高或低优先级全局中断允许位置 1, 来允许中断。如果 $s = 1$, 则影子寄存器 WS、STATUS 和 BSR 的内容将被装入它们对应的寄存器 W、STATUS 和 BSR。如果 $s = 0$, 则不更新这些寄存器 (默认)。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	从堆栈弹出 PC 值, 将 GIEH 或 GIEL 置 1
空操作	空操作	空操作	空操作

示例:

```

RETFIE 1

中断后
PC      = TOS
W       = WS
BSR     = BSR
STATUS  = STATUS
GIE/GIEH, PEIE/GIEL = 1
    
```

RETLW 将立即数返回到 W

语法: RETLW k

操作数: $0 \leq k \leq 255$

操作: $k \rightarrow W$,
(TOS) → PC,
PCLATU 和 PCLATH 保持不变。

受影响的状态位: 无

0000	1100	kkkk	kkkk
------	------	------	------

说明: 将 8 位立即数 k 装入 W。将栈顶内容 (返回地址) 装入程序计数器。高位地址锁存器 (PCLATH) 内容保持不变。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	从堆栈弹出 PC 值, 写入 W
空操作	空操作	空操作	空操作

示例:

```

CALL TABLE ; W contains table
              ; offset value
              ; W now has
              ; table value
:
TABLE
  ADDWF PCL ; W = offset
  RETLW k0 ; Begin table
  RETLW k1 ;
:
:
  RETLW kn ; End of table

执行指令前
W = 07h

执行指令后
W = kn 的值
    
```

RETURN 从子程序返回

语法: RETURN {s}

操作数: $s \in [0,1]$

操作: (TOS) → PC;
如果 $s = 1$
(WS) → W;
(STATUS) → STATUS;
(BSRS) → BSR;
PCLATU 和 PCLATH 保持不变。

受影响的状态位: 无

机器码:

0000	0000	0001	001s
------	------	------	------

说明: 从子程序返回。执行出栈操作, 将栈顶 (TOS) 内容装入程序计数器。如果 $s = 1$, 将影子寄存器 WS、STATUS 和 BSRS 的内容装入它们对应的寄存器, W、STATUS 和 BSR 中。如果 $s = 0$, 则不更新这些寄存器 (默认)。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	从堆栈弹出 PC 值
空操作	空操作	空操作	空操作

示例: RETURN

执行指令后:
PC = TOS

RLCF 对 f 执行带进位的循环左移

语法: RLCF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

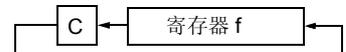
操作: (f<n>) → dest<n+1>,
(f<7>) → C,
(C) → dest<0>

受影响的状态位: C、N 和 Z

机器码:

0011	01da	ffff	ffff
------	------	------	------

说明: 将寄存器 “f” 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: RLCF REG, 0, 0

执行指令前
REG = 1110 0110
C = 0

执行指令后
REG = 1110 0110
W = 1100 1100
C = 1

PIC18F45J10 系列

RLNCF

f 循环左移 (不带进位)

语法: RLNCF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f<n>) \rightarrow \text{dest}<n+1>$,
 $(f<7>) \rightarrow \text{dest}<0>$

受影响的状态位: N 和 Z

机器码:

0100	01da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容循环左移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: RLNCF REG, 1, 0

执行指令前
REG = 1010 1011
执行指令后
REG = 0101 0111

RRCF

对 f 执行带进位的循环右移

语法: RRCF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f<n>) \rightarrow \text{dest}<n-1>$,
 $(f<0>) \rightarrow C$,
 $(C) \rightarrow \text{dest}<7>$

受影响的状态位: C、N 和 Z

机器码:

0011	00da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: RRCF REG, 0, 0

执行指令前
REG = 1110 0110
C = 0
执行指令后
REG = 1110 0110
W = 0111 0011
C = 0

RRNCF **f** 循环右移 (不带进位)

语法: RRNCF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

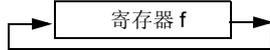
操作: $(f\langle n \rangle) \rightarrow \text{dest}\langle n-1 \rangle$,
 $(f\langle 0 \rangle) \rightarrow \text{dest}\langle 7 \rangle$

受影响的状态位: N 和 Z

机器码:

0100	00da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容循环右移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。如果 a 为 0, 选择快速操作存储区, 忽略 BSR 的值。如果 a 为 1, 使用 BSR 选择存储器 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1: RRNCF REG, 1, 0

执行指令前
REG = 1101 0111

执行指令后
REG = 1110 1011

例 2: RRNCF REG, 0, 0

执行指令前
W = ?
REG = 1101 0111

执行指令后
W = 1110 1011
REG = 1101 0111

SETF 将 f 置为全 1

语法: SETF f{,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: FFh \rightarrow f

受影响的状态位: 无

机器码:

0110	100a	ffff	ffff
------	------	------	------

说明: 将指定寄存器的内容置为 FFh。如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: SETF REG, 1

执行指令前
REG = 5Ah

执行指令后
REG = FFh

PIC18F45J10 系列

SLEEP 进入休眠模式

语法: SLEEP

操作数: 无

操作: 00h → WDT,
0 → WDT 后分频器,
1 → \overline{TO} ,
0 → PD

受影响的状态位: \overline{TO} 和 \overline{PD}

机器码:

0000	0000	0000	0011
------	------	------	------

说明: 掉电状态位 (\overline{PD}) 清零。超时状态位 (\overline{TO}) 置 1。看门狗定时器及其后分频器清零。振荡器停振, 处理器进入休眠模式。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	进入休眠模式

示例: SLEEP

执行指令前
 \overline{TO} = ?
 \overline{PD} = ?

执行指令后
 \overline{TO} = 1 †
 \overline{PD} = 0

† 如果由 WDT 引起唤醒, 则此位将被清零。

SUBFWB W 减去 f (带借位)

语法: SUBFWB f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W) - (f) - (\overline{C}) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0101	01da	ffff	ffff
------	------	------	------

说明:

将 W 的内容减去 f 寄存器的内容和进位 (借位) (通过二进制补码方法进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存入寄存器 f (默认)。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1: SUBFWB REG, 1, 0

执行指令前
REG = 3
W = 2
C = 1

执行指令后
REG = FF
W = 2
C = 0
Z = 0
N = 1 ; 结果为负

例 2: SUBFWB REG, 0, 0

执行指令前
REG = 2
W = 5
C = 1

执行指令后
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; 结果为正

例 3: SUBFWB REG, 1, 0

执行指令前
REG = 1
W = 2
C = 0

执行指令后
REG = 0
W = 2
C = 1
Z = 1 ; 结果为零
N = 0

SUBLW 立即数减去 W 的内容

语法: `SUBLW k`

操作数: $0 \leq k \leq 255$

操作: $k - (W) \rightarrow W$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0000	1000	kkkk	kkkk
------	------	------	------

说明: 用 8 位立即数 k 减去 W。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

例 1: `SUBLW 02h`

执行指令前
W = 01h
C = ?

执行指令后
W = 01h
C = 1 ; 结果为正
Z = 0
N = 0

例 2: `SUBLW 02h`

执行指令前
W = 02h
C = ?

执行指令后
W = 00h
C = 1 ; 结果为零
Z = 1
N = 0

例 3: `SUBLW 02h`

执行指令前
W = 03h
C = ?

执行指令后
W = FFh ; (二进制补码)
C = 0 ; 结果为负
Z = 0
N = 1

SUBWF f 减去 W

语法: `SUBWF f{,d{,a}}`

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) - (W) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0101	11da	ffff	ffff
------	------	------	------

说明:

用寄存器 f 中的内容减去 W 寄存器的内容 (通过二进制补码方式进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

例 1: `SUBWF REG, 1, 0`

执行指令前
REG = 3
W = 2
C = ?

执行指令后
REG = 1
W = 2
C = 1 ; 结果为正
Z = 0
N = 0

例 2: `SUBWF REG, 0, 0`

执行指令前
REG = 2
W = 2
C = ?

执行指令后
REG = 2
W = 0
C = 1 ; 结果为零
Z = 1
N = 0

例 3: `SUBWF REG, 1, 0`

执行指令前
REG = 1
W = 2
C = ?

执行指令后
REG = FFh ; (二进制补码)
W = 2
C = 0 ; 结果为负
Z = 0
N = 1

PIC18F45J10 系列

SUBWFB f 减去 W (带借位)

语法: SUBWFB f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) - (W) - (\bar{C}) \rightarrow dest$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0101	10da	ffff	ffff
------	------	------	------

说明: 将 f 寄存器的内容减去 W 的内容和进位 (借位) (通过二进制补码方法进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存入寄存器 f (默认)。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

例 1: SUBWFB REG, 1, 0

执行指令前
 REG = 19h (0001 1001)
 W = 0Dh (0000 1101)
 C = 1
 执行指令后
 REG = 0Ch (0000 1011)
 W = 0Dh (0000 1101)
 C = 1
 Z = 0
 N = 0 ; 结果为正

例 2: SUBWFB REG, 0, 0

执行指令前
 REG = 1Bh (0001 1011)
 W = 1Ah (0001 1010)
 C = 0
 执行指令后
 REG = 1Bh (0001 1011)
 W = 00h
 C = 1
 Z = 1 ; 结果为零
 N = 0

例 3: SUBWFB REG, 1, 0

执行指令前
 REG = 03h (0000 0011)
 W = 0Eh (0000 1101)
 C = 1
 执行指令后
 REG = F5h (1111 0100)
 : [二进制补码]
 W = 0Eh (0000 1101)
 C = 0
 Z = 0
 N = 1 ; 结果为负

SWAPF 将 f 的高半字节和低半字节交换

语法: SWAPF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f<3:0>) \rightarrow dest<7:4>$,
 $(f<7:4>) \rightarrow dest<3:0>$

受影响的状态位: 无

机器码:

0011	10da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的高半字节和低半字节相交换。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存入寄存器 f (默认)。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: SWAPF REG, 1, 0

执行指令前
 REG = 53h
 执行指令后
 REG = 35h

TBLRD 表读

语法: TBLRD (*; *+; *-; +*)

操作数: 无

操作: 如果执行 TBLRD *,
(程序存储器 (TBLPTR)) → TABLAT ;
TBLPTR 不变
如果执行 TBLRD *+,
(程序存储器 (TBLPTR)) → TABLAT ;
(TBLPTR) + 1 → TBLPTR
如果执行 TBLRD *-,
(程序存储器 (TBLPTR)) → TABLAT ;
(TBLPTR) - 1 → TBLPTR
如果执行 TBLRD +*,
(TBLPTR) + 1 → TBLPTR ;
(程序存储器 (TBLPTR)) → TABLAT

受影响的状态位: 无

机器码:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	---

说明: 该指令用于读取程序存储器 (P.M.) 的内容。使用表指针 (TBLPTR) 对程序存储器进行寻址。
TBLPTR (一个 21 位指针) 指向程序存储器中的某个字节。TBLPTR 的寻址范围为 2 MB。

TBLPTR[0] = 0: 程序存储器字的低有效字节

TBLPTR[0] = 1: 程序存储器字的高有效字节

TBLRD 指令可用如下方法修改 TBLPTR 的值:

- 不变
- 后增
- 后减
- 预增

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作 (读程序存储器)	空操作	空操作 (写 TABLAT)

TBLRD 表读 (续)

例 1: TBLRD *+ ;

执行指令前

TABLAT	=	55h
TBLPTR	=	00A356h
存储单元 (00A356h)	=	34h

执行指令后

TABLAT	=	34h
TBLPTR	=	00A357h

例 2: TBLRD +* ;

执行指令前

TABLAT	=	AAh
TBLPTR	=	01A357h
存储单元 (01A357h)	=	12h
存储单元 (01A358h)	=	34h

执行指令后

TABLAT	=	34h
TBLPTR	=	01A358h

PIC18F45J10 系列

TBLWT 表写

语法: TBLWT (*; *+; *-; +*)

操作数: 无

操作: 如果执行 TBLWT*,
(TABLAT) → 保持寄存器;
TBLPTR 不变
如果执行 TBLWT*+,
(TABLAT) → 保持寄存器;
(TBLPTR) + 1 → TBLPTR
如果执行 TBLWT*-,
(TABLAT) → 保持寄存器;
(TBLPTR) - 1 → TBLPTR
如果执行 TBLWT*+*,
(TBLPTR) + 1 → TBLPTR ;
(TABLAT) → 保持寄存器

受影响的状态位: 无

机器码:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	---

说明: 此指令使用 TBLPTR 的低 3 位来确定要将 TABLAT 中的内容写入 8 个保持寄存器中的哪一个。该保持寄存器用于对程序存储器 (P.M.) 的内容编程。(关于对闪存存储器编程的更多详情, 请参见第 6.0 节 “闪存程序存储器”。)

TBLPTR (一个 21 位指针) 指向程序存储器中的某个字节。TBLPTR 的寻址范围为 2 MB。TBLPTR 的 LSb 选择要访问的程序存储器单元。

TBLPTR[0] = 0: 程序存储器字的低有效字节

TBLPTR[0] = 1: 程序存储器字的高有效字节

TBLWT 指令可用如下方法修改 TBLPTR 的值:

- 不变
- 后增
- 后减
- 预增

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作 (读 TABLAT)	空操作	空操作 (写保持 寄存器)

TBLWT 表写 (续)

例 1: TBLWT *+;

执行指令前

TABLAT	=	55h
TBLPTR	=	00A356h
保持寄存器 (00A356h)	=	FFh

执行指令后 (写表操作完成)

TABLAT	=	55h
TBLPTR	=	00A357h
保持寄存器 (00A356h)	=	55h

例 2: TBLWT +*;

执行指令前

TABLAT	=	34h
TBLPTR	=	01389Ah
保持寄存器 (01389Ah)	=	FFh
保持寄存器 (01389Bh)	=	FFh

执行指令后 (写表操作完成)

TABLAT	=	34h
TBLPTR	=	01389Bh
保持寄存器 (01389Ah)	=	FFh
保持寄存器 (01389Bh)	=	34h

TSTFSZ 测试 f, 为 0 则跳过

语法: TSTFSZ f {,a}

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: f 为 0 则跳过

受影响的状态位: 无

机器码:

0110	011a	ffff	ffff
------	------	------	------

说明: 如果 $f = 0$, 丢弃已取的指令, 转而执行一条 NOP 指令, 使这条指令成为双周期指令。
 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
 如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1(2)

注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

执行指令前
 PC = 地址 (HERE)

执行指令后
 如果 CNT = 00h,
 PC = Address (ZERO)
 如果 CNT \neq 00h,
 PC = 地址 (NZERO)

XORLW 立即数与 w 作逻辑异或运算

语法: XORLW k

操作数: $0 \leq k \leq 255$

操作: (W) .XOR. k \rightarrow W

受影响的状态位: N 和 Z

机器码:

0000	1010	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与 8 位立即数 k 作逻辑异或运算。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

示例: XORLW 0AFh

执行指令前
 W = B5h

执行指令后
 W = 1Ah

PIC18F45J10 系列

XORWF W 与 f 作逻辑异或运算

语法: XORWF f{,d{,a}}

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: (W) .XOR. (f) → dest

受影响的状态位: N 和 Z

机器码:

0001	10da	ffff	ffff
------	------	------	------

说明: 将 W 的内容与寄存器 f 的内容作逻辑异或运算。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。
如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。
如果 a 为 0 且使能了扩展的指令集, 只要 $f \leq 95$ (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 21.2.3 节“立即数变址寻址模式中面向字节和面向位的指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: XORWF REG, 1, 0

执行指令前
REG = AFh
W = B5h
执行指令后
REG = 1Ah
W = B5h

21.2 扩展的指令集

除了PIC18指令集的75条标准指令之外，PIC18F45J10系列器件还提供针对内核CPU功能的可选扩展指令。这些新增的功能包括8条额外的指令，它们增强了间接和变址寻址操作以及对许多标准PIC18指令执行立即数变址寻址模式的功能。

这些扩展指令集的额外功能在默认情况下是禁止的。用户必须通过将XINST配置位置1，才能启用它们。

扩展指令集中的指令可以全部被归为立即数操作类指令，它们既可以控制文件选择寄存器也可以使用这些寄存器进行变址寻址。ADDFSR和SUBFSR有两条对应的特殊指令ADDULNK和SUBULNK，可以直接对FSR2进行操作，并允许在执行后自动返回。

这些扩展的指令专门用于优化用高级语言，特别是C语言编写的可重入程序代码（也就是递归调用或使用软件堆栈的代码）。此外，它们使用户能更有效地用高级语言对数据结构执行特定的操作。包括：

- 在进入和退出子程序时对软件堆栈空间进行动态分配和释放
- 功能指针调用
- 对软件堆栈指针进行控制
- 对软件堆栈中的变量进行控制

表 21-3 提供了扩展指令集中的指令汇总。第 21.2.2 节“扩展指令集”中提供了详细的说明。表 21-1（第 242 页）提供了适用于标准和扩展的 PIC18 指令集的操作码字段的说明。

注： 扩展的指令集和立即数变址寻址模式是专为优化用 C 语言编写的应用程序而设计的，用户可能永远不会在汇编器中直接使用这些指令。对于那些查看由编译器生成的代码的用户，这些命令的语法可作为参考。

21.2.1 扩展指令的语法

大部分扩展指令都使用变址参数，同时使用一个文件选择寄存器和某一偏移量来指定源寄存器或目标寄存器。当指令的参数作为变址寻址的一部分时，会用方括号（“[]”）把它括起来。这时表示此参数用作变址地址或偏移量。如果 MPASM™ 汇编器判断出一个变址地址或偏移量没有被括起来，它就会给出错误报告。

当启用扩展的指令集时，方括号也用于表示面向字节和面向位的指令中的变址参数。这是对标准指令语法的另一处更改。更多详情，请参见第 21.2.3.1 节“标准 PIC18 命令的扩展指令语法”。

注： 以前，在 PIC18 和更早的指令集中使用方括号来表示可选参数。在文本和以后的文档中，可选参数将用大括号（“{}”）表示。

表 21-3: PIC18 指令集的扩展

助记符, 操作数	说明	周期数	16 位指令字		受影响的状态位	
			MSb	LSb		
ADDFSR f, k	将立即数与 FSR 相加	1	1110	1000	ffkk kkkk	无
ADDULNK k	将立即数与 FSR2 相加并返回	2	1110	1000	11kk kkkk	无
CALLW	使用 WREG 调用子程序	2	0000	0000	0001 0100	无
MOVSF z_s, f_d	将 z_s (源) 地址装入第一个字 将 f_d (目标) 地址装入第二个字	2	1110	1011	0zzz zzzz 1111 ffff ffff ffff	无
MOVSS z_s, z_d	将 z_s (源) 地址装入第一个字 将 z_d (目标) 地址装入第二个字	2	1110	1011	1zzz zzzz 1111 xxxx xzzz zzzz	无
PUSHL k	将立即数保存在 FSR2, FSR2 减 1	1	1110	1010	kkkk kkkk	无
SUBFSR f, k	FSR 减去立即数	1	1110	1001	ffkk kkkk	无
SUBULNK k	FSR2 减去立即数并返回	2	1110	1001	11kk kkkk	无

CALLW 使用 WREG 调用子程序

语法:	CALLW				
操作数:	无				
操作:	(PC+2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">0001</td> <td style="padding: 2px;">0100</td> </tr> </table>	0000	0000	0001	0100
0000	0000	0001	0100		

说明
首先, 返回地址 (PC+2) 被压入返回堆栈。接下来, 将 W 寄存器的内容写入 PCL, PCL 现有的值被丢弃。然后, PCLATH 和 PCLATU 的内容被分别锁存到 PCH 和 PCU。第二个周期执行一条 NOP 指令, 并同时取出下一条新的指令。
和 CALL 指令不一样, 该指令没有更新 W、STATUS 或 BSR 寄存器的选项。

指令字数: 1
指令周期数: 2
Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 WREG	将 PC 压入堆栈	空操作
空操作	空操作	空操作	空操作

示例: HERE CALLW

执行指令前
PC = 地址 (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h

执行指令后
PC = 001006h
TOS = 地址 (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

MOVSF 将变址寻址单元的内容传送到 f

语法:	MOVSF [z _s], f _d								
操作数:	0 ≤ z _s ≤ 127 0 ≤ f _d ≤ 4095								
操作:	((FSR2) + z _s) → f _d								
受影响的状态位:	无								
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">1110</td> <td style="padding: 2px;">1011</td> <td style="padding: 2px;">0zzzz</td> <td style="padding: 2px;">zzzzz_s</td> </tr> <tr> <td style="padding: 2px;">1111</td> <td style="padding: 2px;">ffff</td> <td style="padding: 2px;">ffff</td> <td style="padding: 2px;">fffff_d</td> </tr> </table>	1110	1011	0zzzz	zzzzz _s	1111	ffff	ffff	fffff _d
1110	1011	0zzzz	zzzzz _s						
1111	ffff	ffff	fffff _d						

说明
将源寄存器的内容传送到目标寄存器 f_d。通过将第一个字中的 7 位立即数偏移量 z_s 与 FSR2 的值相加来确定源寄存器的实际地址。第二个字中的 12 位立即数 f_d 指向目标寄存器的地址。两个地址均可以处于 4096 字节的数据空间 (000h 到 FFFh) 中的任何位置。
MOVSF 指令中的目标寄存器不能是 PCL、TOSU、TOSH 或 TOSL。
如果计算得到的源地址指向间接寻址寄存器, 将返回 00h。

指令字数: 2
指令周期数: 2
Q 周期操作:

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	空操作 非虚拟读取	空操作	写目标寄存器 f

示例: MOVSF [05h], REG2

执行指令前
FSR2 = 80h
85h 单元的
内容 = 33h
REG2 = 11h

执行指令后
FSR2 = 80h
85h 单元的
内容 = 33h
REG2 = 33h

PIC18F45J10 系列

MOVSS 变址寻址数据传送

语法: MOVSS [z_s], [z_d]
 操作数: 0 ≤ z_s ≤ 127
 0 ≤ z_d ≤ 127
 操作: ((FSR2) + z_s) → ((FSR2) + z_d)

受影响的状态位: 无

机器码:

第一个字 (源)	1110	1011	1zzz	zzzz _s
第二个字 (目标)	1111	xxxx	xzzz	zzzz _d

说明
 将源寄存器的内容传送到目标寄存器。通过将 FSR2 中的值分别加上 7 位立即数偏移量 z_s 或 z_d 来确定源寄存器和目标寄存器的地址。两个寄存器可以是 4096 字节数据存储空间 (000h 到 FFFh) 中的任意单元。
 MOVSS 指令不能使用 PCL、TOS、TOSH 或 TOSL 作为目标寄存器。如果计算得到的源地址指向间接寻址寄存器，将返回 00h。如果计算得到的目标地址指向间接寻址寄存器，将执行一条 NOP 指令。

指令字数: 2
 指令周期数: 2
 Q 周期操作:

	Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	确定源地址	读源寄存器
译码	确定目标地址	确定目标地址	确定目标地址	写目标寄存器

示例: MOVSS [05h], [06h]

执行指令前
 FSR2 = 80h
 85h 单元的内容 = 33h
 86h 单元的内容 = 11h
 执行指令后
 FSR2 = 80h
 85h 单元的内容 = 33h
 86h 单元的内容 = 33h

PUSHL 将立即数保存在 FSR2 所指的单元, FSR2 减 1

语法: PUSHL k
 操作数: 0 ≤ k ≤ 255
 操作: k → (FSR2),
 FSR2 - 1 → FSR2

受影响的状态位: 无

机器码:

1111	1010	kkkk	kkkk
------	------	------	------

说明: 8 位立即数 k 被写入由 FSR2 指定的数据存储单元。操作完成后 FSR2 减 1。此指令允许用户将值压入软件堆栈。

指令字数: 1
 指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例: PUSHL 08h

执行指令前
 FSR2H:FSR2L = 01ECh
 存储单元 (01ECh) = 00h
 执行指令后
 FSR2H:FSR2L = 01EBh
 存储单元 (01ECh) = 08h

PIC18F45J10 系列

21.2.3 立即数变址寻址模式中面向字节和面向位的指令

注： 使能 PIC18 扩展指令集可能导致常规应用程序运行不正常或完全失败。

使能扩展的指令集除了使能了其中的 8 条新命令之外，还使能了立即数变址寻址模式（第 5.5.1 节“使用立即数偏移量进行变址寻址”）。这将导致标准 PIC18 指令的地址解析方法有很大变化。

当禁用扩展的指令集时，嵌入在操作码中的地址被视为立即数存储单元：可以是快速操作存储区中的单元（ $a = 0$ ），或由 BSR 指定的 GPR 存储区中的单元（ $a = 1$ ）。当使能扩展的指令集且 $a = 0$ 时，地址为 5Fh 或以下的文件寄存器参数被解析为 FSR2 中的指针值的偏移量，而不是一个立即数地址。对于实际应用来说，这意味着所有使用快速操作 RAM 位作为参数的指令，即所有面向字节或位的指令，或者几乎半数的 PIC18 内核指令，在使能了扩展的指令集时操作都会有所不同。

当 FSR2 的内容为 00h 时，快速操作 RAM 的边界会被重新映射到它们的原始值。这对于编写向后兼容的代码很有用处。如果使用此技术，有必要在“C”程序调用汇编子程序时保存 FSR2 的值，并在返回时将它恢复，这样做的目的是保存堆栈指针。用户还必须记住扩展指令集的语法要求（见第 21.2.3.1 节“标准 PIC18 命令的扩展指令语法”）。

虽然立即数变址寻址模式对于动态堆栈和指针控制很有用处，但是如果不小心误用了寄存器，也会非常麻烦。已经习惯使用 PIC18 编程的用户必须记住，在使能了扩展的指令集后，地址小于或等于 5Fh 的寄存器用于立即数变址寻址。

下面是在立即数变址寻址模式中，一些面向字节和位的指令的示例，通过示例可以看出指令执行如何受到影响。示例中的操作数条件适用于所有指令。

21.2.3.1 标准 PIC18 命令的扩展指令语法

当使能了扩展的指令集时，立即数偏移量 k 被用来替换标准的面向字节和位的命令中的文件寄存器参数 f 。如前所述，只有在“ f ”小于或等于 5Fh 时才会发生这种情况。当使用偏移量时，该偏移量必须用方括号“ $[\]$ ”标出。因为在扩展的指令集中，将方括号中的数值解析为变址地址或偏移量。省略括号，或在括号内使用大于 5Fh 的值会在 MPASM 汇编器中产生错误。

如果变址参数已被加上了括号，那么就不再需要指定快速操作 RAM 参数；此参数被假定为“0”。这与标准操作（禁止扩展的指令集时）正好相反。在变址寻址模式中，定义快速操作 RAM 位也将在 MPASM 汇编器中产生错误。

目标参数“ d ”的操作和以前一样。

如需获取有关使能扩展的指令集的支持信息，请参见 MPLAB® IDE、MPASM™ 或 MPLAB C18 文档。

21.2.4 使能扩展的指令集时的注意事项

需要注意的是并非所有用户都有必要使用扩展的指令集，尤其是那些不使用软件堆栈的用户。

此外，立即数变址寻址模式可能会对写入 PIC18 汇编器的常规应用程序带来问题。这是因为常规的指令会尝试寻址快速操作存储区中地址低于 5Fh 的寄存器。当使能了扩展的指令集时，这些地址被解析为相对于 FSR2 的立即数偏移量，所以应用程序会读或写错误的地址。

将应用程序移植到 PIC18F45J10 系列器件时，代码类型是非常重要的。在使用扩展的指令集时，用 C 语言编写的代码较长的可重入应用程序会运行的很好，而大量使用快速操作存储区的常规应用程序不会获得任何益处。

ADDWF 将 W 与变址寻址单元的内容相加 (立即数变址寻址模式)

语法: ADDWF [k],d

操作数: $0 \leq k \leq 95$
 $d \in [0,1]$

操作: $(W) + ((FSR2) + k) \rightarrow dest$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0010	01d0	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容与由 FSR2 加上偏移量 k 指定的寄存器的内容相加。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例: ADDWF [OFST], 0

执行指令前

W	=	17h
OFST	=	2Ch
FSR2	=	0A00h
0A2Ch 单元的内容	=	20h

执行指令后

W	=	37h
0A2Ch 单元的内容	=	20h

BSF 将变址寻址单元相应位置 1 (立即数变址寻址模式)

语法: BSF [k], b

操作数: $0 \leq f \leq 95$
 $0 \leq b \leq 7$

操作: $1 \rightarrow ((FSR2) + k) $

受影响的状态位: 无

机器码:

1000	bbb0	kkkk	kkkk
------	------	------	------

说明: 将由 FSR2 加上偏移量 k 指定的寄存器中的位 b 置 1。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: BSF [FLAG_OFST], 7

执行指令前

FLAG_OFST	=	0Ah
FSR2	=	0A00h
0A0Ah 单元的内容	=	55h

执行指令后

0A0Ah 单元的内容	=	D5h
-------------	---	-----

SETF 将变址寻址单元置全 1 (立即数变址寻址模式)

语法: SETF [k]

操作数: $0 \leq k \leq 95$

操作: $FFh \rightarrow ((FSR2) + k)$

受影响的状态位: 无

机器码:

0110	1000	kkkk	kkkk
------	------	------	------

说明: 将由 FSR2 加上偏移量 k 指定的寄存器的内容置为 FFh。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写寄存器

示例: SETF [OFST]

执行指令前

OFST	=	2Ch
FSR2	=	0A00h
0A2Ch 单元的内容	=	00h

执行指令后

0A2Ch 单元的内容	=	FFh
-------------	---	-----

PIC18F45J10 系列

21.2.5 使用 MICROCHIP MPLAB® IDE 工具的注意事项

最新版本的 Microchip 软件工具，可以完全支持 PIC18F45J10 系列器件的扩展指令集。包括 MPLAB C18 C 编译器、MPASM 汇编语言和 MPLAB 集成开发环境 (IDE)。

在选择了使用软件开发的目标器件后，MPLAB IDE 将该器件的默认配置位自动置 1。在禁用扩展指令集和立即数变址寻址模式时，XINST 配置位的默认设置是 0。在编程过程中必须将 XINST 位置 1 才能正确地利用扩展指令集开发应用程序。

要使用扩展的指令集开发软件，用户必须设置他们的语言工具以实现扩展指令和变址寻址模式的支持。根据所使用的环境，可以通过以下几种方法：

- 开发环境中的菜单选项或对话框，允许用户配置项目的语言工具及其设置
- 命令行选项
- 源代码中的伪指令

这些选项在不同的编译器、汇编器和开发环境中将有所不同。建议用户在其开发系统所附带的文档中查询相关的信息。

22.0 开发支持

一系列硬件及软件开发工具对 PICmicro® 单片机提供支持：

- 集成开发环境
 - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
 - MPASM™ 汇编器
 - MPLAB C18 和 MPLAB C30 C 编译器
 - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
 - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
 - MPLAB SIM 软件模拟器
- 仿真器
 - MPLAB ICE 2000 在线仿真器
 - MPLAB ICE 4000 在线仿真器
- 在线调试器
 - MPLAB ICD 2
- 器件编程器
 - PICSTART® Plus 开发编程器
 - MPLAB PM3 器件编程器
- 低成本演示和开发板及评估工具包

22.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
 - 模拟器
 - 编程器（单独销售）
 - 仿真器（单独销售）
 - 在线调试器（单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（汇编语言或 C 语言）
- 点击一次即可完成汇编（或编译）并将代码下载到 PICmicro MCU 仿真器和模拟器工具中（自动更新所有项目信息）
- 可使用如下各项进行调试：
 - 源文件（汇编语言或 C 语言）
 - 混合汇编语言和 C 语言
 - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

22.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PICmicro MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

22.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 系列单片机和 dsPIC30F 系列数据信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

22.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用中。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

22.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

22.6 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器在指令级对 PICmicro MCU 和 dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使模拟器还能记录和跟踪程序的执行、I/O 的动作以及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

22.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PICmicro 单片机的设计工具。MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PICmicro 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

22.8 MPLAB ICE 4000 高性能在线仿真器

MPLAB ICE 4000 在线仿真器旨在为产品开发工程师提供一整套用于高端 PICmicro MCU 和 dsPIC DSC 的设计工具。MPLAB ICE 4000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 4000 是高级的仿真系统，除具备 MPLAB ICE 2000 的所有功能外，它还增加了适用于 dsPIC30F 和 PIC18XXXX 器件的仿真存储容量以及高速性能。该仿真器的先进特性包括复杂触发和定时功能及高达 2 Mb 的仿真存储容量。

MPLAB ICE 4000 在线仿真系统设计为一款实时仿真系统，该仿真系统具备通常只有在更加昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft Windows 32 位操作系统可使这些功能在一个简单而统一的应用程序中得以很好的利用。

22.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PICmicro MCU，可用于开发本系列及其他 PICmicro MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程 (In-Circuit Serial Programming™, ICSP™) 协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。MPLAB ICD 2 还可用作某些 PICmicro 器件的开发编程器。

22.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PICmicro 器件进行读取、验证和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

PIC18F45J10 系列

22.11 PICSTART Plus 开发编程器

PICSTART Plus 开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM (RS-232) 端口与 PC 相连。MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PICmicro 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

22.12 演示、开发和评估板

有许多演示、开发和评估板可用于各种 PICmicro MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于测试和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart® 电池管理、SEEVAL® 评估系统、 Σ - Δ ADC、流速传感器，等等。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 (www.microchip.com) 以及最新的 “*Product Selector Guide (产品选型指南)*” (DS00148)。

注:

PIC18F45J10 系列

注:

23.0 电气规范

绝对最大值 (†)

偏置电压下的环境温度	-40°C 至 +125°C
储存温度	-65°C 至 +150°C
任一数字 I/O 引脚 (除了 VDD 和 $\overline{\text{MCLR}}$ 之外) 相对于 VSS 的电压	-0.3V 至 5.5V
任一数模组合引脚 (除了 VDD 和 $\overline{\text{MCLR}}$ 之外) 相对于 VSS 的电压	-0.3V 至 (VDD + 0.3V)
VDDCORE 相对于 VSS 的电压	-0.3V 至 2.75V
VDD 相对于 VSS 的电压	-0.3V 至 3.6V
$\overline{\text{MCLR}}$ 相对于 VSS 的电压	0V 至 5.5V
总功耗 (注 1)	1.0W
VSS 引脚的最大输出电流	300 mA
VDD 引脚的最大输入电流	250 mA
输入箝位电流 I _{IK} (V _I < 0 或 V _I > VDD)	TBD
输出箝位电流 I _{OK} (V _O < 0 或 V _O > VDD)	TBD
PORTB 和 PORTC I/O 引脚的最大输出灌电流	25 mA
PORTA、PORTD 和 PORTE I/O 引脚的最大输出灌电流	2 mA
PORTB 和 PORTC I/O 引脚的最大输出拉电流	25 mA
PORTA、PORTD 和 PORTE I/O 引脚的最大输出拉电流	2 mA
所有端口的最大灌电流	200 mA
所有端口的最大拉电流	200 mA

注 1: 功耗按如下公式计算:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

† 注意: 如果器件的工作条件超过“绝对最大值”列出的范围, 就可能会对器件造成永久性损坏。上述值仅为运行条件极大值, 我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大额定值条件下, 其稳定性会受到影响。

PIC18F45J10 系列

图 23-1: PIC18LF45J10 系列电压—频率关系图 (工业级)

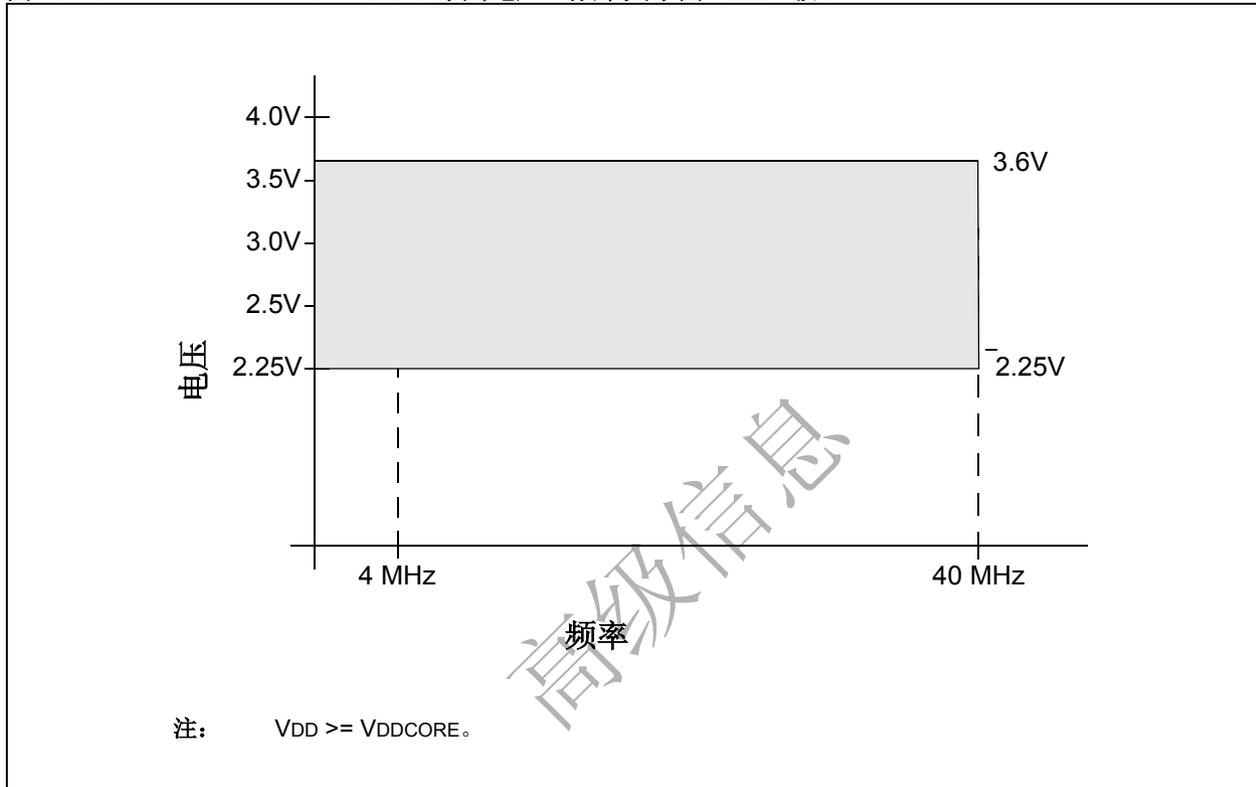
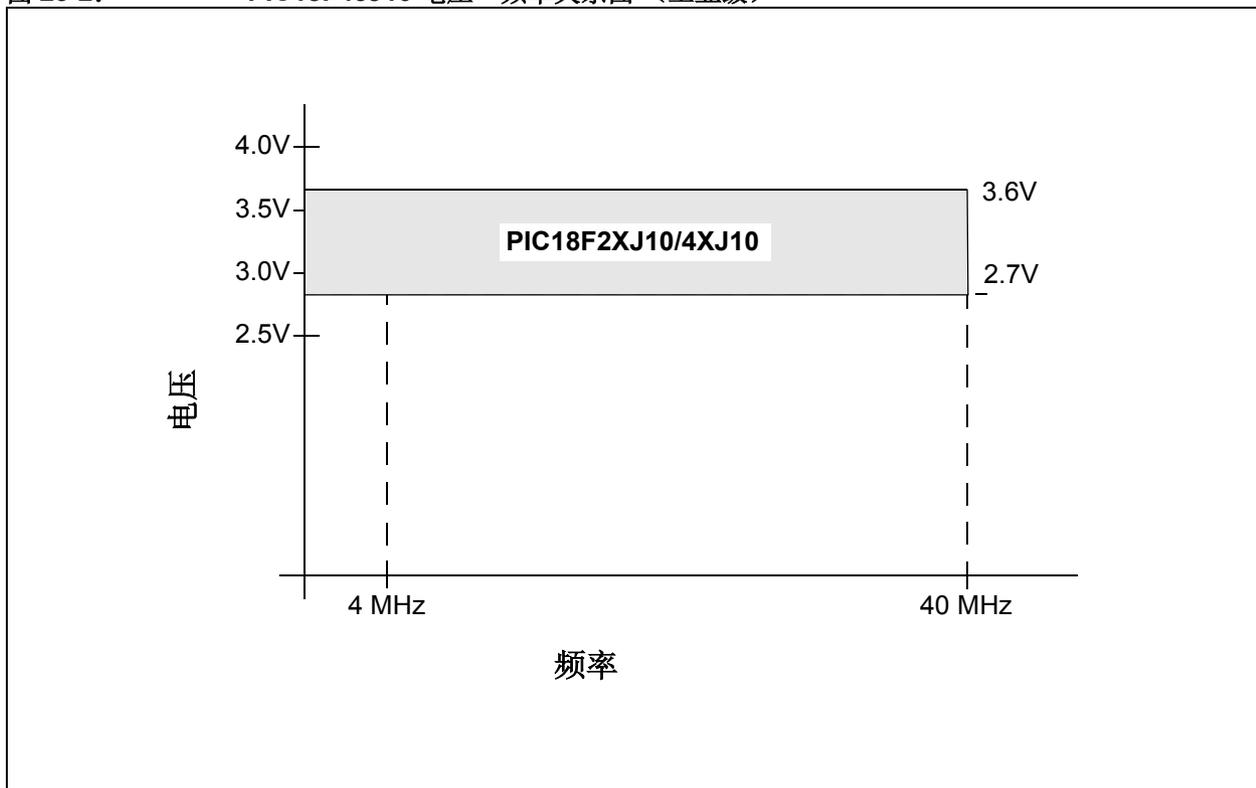


图 23-2: PIC18F45J10 电压—频率关系图 (工业级)



23.1 直流规范:

供电电压

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$ (工业级)					
参数 编号	符号	特性	最小值	典型 值	最大 值	单位	条件
D001	VDD	供电电压	VDDCORE ⁽²⁾	—	3.6	V	
D001B	VDDCORE	单片机内核的外部电源	2.0	—	2.75	V	仅在标有“LF”的器件上有效。 如需详细信息,请参见第 20.3 节 “片内稳压器”。
D002	VDR	RAM 数据保存电压 ⁽¹⁾	1.5	—	—	V	
D003	VPOR	VDD 启动电压 确保能够产生内部 上电复位信号	—	—	TBD	V	如需详细信息,请参见第 4.3 节 “上电复位 (POR)”。
D004	SVDD	VDD 上升速率 确保能够产生内部 上电复位信号	0.05	—	—	V/ms	如需详细信息,请参见第 4.3 节 “上电复位 (POR)”。
D005A	VBOR	欠压复位电压	—	—	2.6	V	使能了片内稳压器。如需详细信 息,请参见第 4.3 节“上电复位 (POR)”。

图注: TBD = 待定

注 1: 该电压是休眠模式或器件复位状态下在不丢失 RAM 数据的前提下的最小 VDD。

2: 标有“F”的器件中, $V_{DD} \geq 2.7V$ 。

PIC18F45J10 系列

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)			
参数 编号	器件	典型 值	最大 值	单位	条件
掉电电流 (IPD) ⁽¹⁾					
	所有器件	TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$
	所有器件	TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$

图注: TBD = 待定

- 注
- 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS, 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
 - 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 IDD 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR = VDD; 根据具体应用使能或禁止 WDT。
 - 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 k Ω 。
 - 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数 编号	器件	典型 值	最大 值	单位	条件		
供电电流 (IDD) (2,3)							
	所有器件	TBD	TBD	μA	-40°C	VDD = 2.5V FOSC = 31 kHz (RC_RUN 模式, 内部振荡器作为时钟源)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C		VDD = 3.3V
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C	VDD = 2.5V FOSC = 31 kHz (RC_IDLE 模式, 内部振荡器作为时钟源)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
所有器件	TBD	TBD	μA	-40°C	VDD = 3.3V		
	TBD	TBD	μA	$+25^{\circ}\text{C}$			
	TBD	TBD	μA	$+85^{\circ}\text{C}$			

图注: TBD = 待定

- 注 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS, 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 IDD 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR = VDD; 根据具体应用使能或禁止 WDT。
- 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

PIC18F45J10 系列

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数 编号	器件	典型 值	最大 值	单位	条件		
供电电流 (IDD) (2,3)							
	所有器件	TBD	TBD	μA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 1 MHz (PRI_RUN 模式, EC 振荡器)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	mA	-40°C		$V_{DD} = 3.3\text{V}$
		TBD	TBD	mA	$+25^{\circ}\text{C}$		
		TBD	TBD	mA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	mA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 4 MHz (PRI_RUN 模式, EC 振荡器)	
		TBD	TBD	mA	$+25^{\circ}\text{C}$		
		TBD	TBD	mA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	mA	-40°C		$V_{DD} = 3.3\text{V}$
		TBD	TBD	mA	$+25^{\circ}\text{C}$		
		TBD	TBD	mA	$+85^{\circ}\text{C}$		
所有器件	TBD	TBD	mA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 40 MHz (PRI_RUN 模式, EC 振荡器)		
	TBD	TBD	mA	$+25^{\circ}\text{C}$			
	TBD	TBD	mA	$+85^{\circ}\text{C}$			
所有器件	TBD	TBD	mA	-40°C		$V_{DD} = 3.3\text{V}$	
	TBD	TBD	mA	$+25^{\circ}\text{C}$			
	TBD	TBD	mA	$+85^{\circ}\text{C}$			

图注: TBD = 待定

- 注 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或者 V_{SS} , 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 I_{DD} 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD} ;
MCLR = V_{DD} ; 根据具体应用使能或禁止 WDT。
- 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 $k\Omega$ 。
- 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数 编号	器件	典型 值	最大 值	单位	条件	
供电电流 (IDD) (2)						
	所有器件	TBD	TBD	mA	-40°C	VDD = 2.5V FOSC = 4 MHz, (PRI_RUN 模式)
		TBD	TBD	mA	$+25^{\circ}\text{C}$	
		TBD	TBD	mA	$+85^{\circ}\text{C}$	
	所有器件	TBD	TBD	mA	-40°C	VDD = 3.3V FOSC = 4 MHz, (PRI_RUN 模式)
		TBD	TBD	mA	$+25^{\circ}\text{C}$	
		TBD	TBD	mA	$+85^{\circ}\text{C}$	
	所有器件	TBD	TBD	mA	-40°C	VDD = 2.5V FOSC = 10 MHz, (PRI_RUN 模式)
		TBD	TBD	mA	$+25^{\circ}\text{C}$	
		TBD	TBD	mA	$+85^{\circ}\text{C}$	
	所有器件	TBD	TBD	mA	-40°C	VDD = 3.3V FOSC = 10 MHz, (PRI_RUN 模式)
		TBD	TBD	mA	$+25^{\circ}\text{C}$	
		TBD	TBD	mA	$+85^{\circ}\text{C}$	

图注: TBD = 待定

- 注 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS, 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 IDD 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR = VDD; 根据具体应用使能或禁止 WDT。
- 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

PIC18F45J10 系列

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数 编号	器件	典型 值	最大 值	单位	条件		
供电电流 (IDD) (2,3)							
	所有器件	TBD	TBD	μA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 1 MHz (PRI_IDLE 模式, EC 振荡器)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C		$V_{DD} = 3.3\text{V}$
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 4 MHz (PRI_IDLE 模式, EC 振荡器)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	mA	-40°C		$V_{DD} = 3.3\text{V}$
		TBD	TBD	mA	$+25^{\circ}\text{C}$		
		TBD	TBD	mA	$+85^{\circ}\text{C}$		
所有器件	TBD	TBD	mA	-40°C	$V_{DD} = 2.5\text{V}$ Fosc = 40 MHz (PRI_IDLE 模式, EC 振荡器)		
	TBD	TBD	mA	$+25^{\circ}\text{C}$			
	TBD	TBD	mA	$+85^{\circ}\text{C}$			
所有器件	TBD	TBD	mA	-40°C		$V_{DD} = 3.3\text{V}$	
	TBD	TBD	mA	$+25^{\circ}\text{C}$			
	TBD	TBD	mA	$+85^{\circ}\text{C}$			

图注: TBD = 待定

- 注
- 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或者 V_{SS} , 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
 - 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 I_{DD} 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD} ;
MCLR = V_{DD} ; 根据具体应用使能或禁止 WDT。
 - 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 $k\Omega$ 。
 - 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数 编号	器件	典型 值	最大 值	单位	条件		
供电电流 (IDD) (2,3)							
	所有器件	TBD	TBD	μA	-40°C	VDD = 2.5V Fosc = 32 kHz ⁽⁴⁾ (SEC_RUN 模式, Timer1 作为时钟源)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C		VDD = 3.3V
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
	所有器件	TBD	TBD	μA	-40°C	VDD = 2.5V Fosc = 32 kHz ⁽⁴⁾ (SEC_IDLE 模式, Timer1 作为时钟源)	
		TBD	TBD	μA	$+25^{\circ}\text{C}$		
		TBD	TBD	μA	$+85^{\circ}\text{C}$		
所有器件	TBD	TBD	μA	-40°C	VDD = 3.3V		
	TBD	TBD	μA	$+25^{\circ}\text{C}$			
	TBD	TBD	μA	$+85^{\circ}\text{C}$			

图注: TBD = 待定

- 注 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS, 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 IDD 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR = VDD; 根据具体应用使能或禁止 WDT。
- 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

PIC18F45J10 系列

23.2 直流规范:

掉电和供电电流

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

PIC18F45J10 系列 (工业级)		标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)			
参数 编号	器件	典型 值	最大 值	单位	条件
D022 (ΔI_{WDT})	模块差分电流 (ΔI_{WDT} 、 ΔI_{OSCB} 和 ΔI_{AD}) 看门狗定时器	TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$
		TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$
D025 (ΔI_{OSCB})	Timer1 振荡器	TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$
		TBD	TBD	μA	-40°C
		TBD	TBD	μA	$+25^{\circ}\text{C}$
		TBD	TBD	μA	$+85^{\circ}\text{C}$
D026 (ΔI_{AD})	A/D 转换器	TBD	TBD	μA	-40°C 至 $+85^{\circ}\text{C}$
		TBD	TBD	μA	-40°C 至 $+85^{\circ}\text{C}$

图注: TBD = 待定

- 注
- 1: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS, 以及禁止了所有会带来新增电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
 - 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下, 所有 I_{DD} 测量的测试条件为:
OSC1 = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR = VDD; 根据具体应用使能或禁止 WDT。
 - 3: 当器件振荡器配置为 RC 模式时, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 来估算, 其中 REXT 的单位是 k Ω 。
 - 4: 标准低成本 32 kHz 晶振的工作温度范围从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

23.3 直流规范:

PIC18F24J10/25J10/44J10/45J10 (工业级)

PIC18LF24J10/25J10/44J10/45J10 (工业级)

直流规范			标准工作条件 (除非另行声明)					
			工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数编号	符号	特性	最小值	最大值	单位	条件		
D030	V _{IL}	输入低电压 I/O 端口: 带 TTL 缓冲器	V _{SS}	0.15 V _{DD}	V	V _{DD} < 3.3V 3.3V ≤ V _{DD} ≤ 3.6V		
D030A				0.8	V			
D031			带施密特触发缓冲器	V _{SS}	0.2 V _{DD}		V	
D032			MCLR	V _{SS}	0.2 V _{DD}		V	
D033			OSC1	V _{SS}	0.3 V _{DD}		V	HS 和 HSPLL 模式 EC 模式 ⁽¹⁾
D033A			OSC1	V _{SS}	0.2 V _{DD}		V	
D034			T1CKI	V _{SS}	0.3		V	
D040	V _{IH}	输入高电压 I/O 端口: 带 TTL 缓冲器	0.25 V _{DD} + 0.8V	V _{DD}	V	V _{DD} < 3.3V 3.3V ≤ V _{DD} ≤ 3.6V		
D040A				2.0	V _{DD}		V	
D041			带施密特触发缓冲器	0.8 V _{DD}	V _{DD}		V	
D042			MCLR	0.8 V _{DD}	V _{DD}		V	
D043			OSC1	0.7 V _{DD}	V _{DD}		V	HS 和 HSPLL 模式 EC 模式
D043A			OSC1	0.8 V _{DD}	V _{DD}		V	
D044			T1CKI	TBD	V _{DD}		V	
D060	I _{IL}	输入泄漏电流 ^(2,3) I/O 端口	—	TBD	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} 引脚处于高阻态		
D061			MCLR	—	TBD		μA	
D063			OSC1	—	TBD		μA	
D070	IPU IPURB	弱上拉电流 PORTB 弱上拉电流	50	400	μA	V _{DD} = 3.3V, V _{PIN} = V _{SS}		

图注: TBD = 待定

- 注 1: 在 RC 振荡器配置中, OSC1/CLKI 引脚被配置为施密特触发器输入。在 RC 模式中建议不要使用外部时钟驱动 PICmicro[®] 器件。
- 2: MCLR 引脚上的泄漏电流主要取决于施加的电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。
- 3: 负电流定义为引脚输出的电流。

PIC18F45J10 系列

23.3 直流规范: PIC18F24J10/25J10/44J10/45J10 (工业级) PIC18LF24J10/25J10/44J10/45J10 (工业级) (续)

直流规范			标准工作条件 (除非另行声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)			
参数编号	符号	特性	最小值	最大值	单位	条件
D080A	VOL	输出低电压 I/O 端口	—	TBD	V	TBD, $V_{DD} = 3.3\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D080B			—	—		
D080C			—	—		
D083		OSC2/CLKO (EC 模式)	—	0.6	V	$I_{OL} = 1.6\text{ mA}$, $V_{DD} = 3.3\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D090A	VOH	输出高电压 ⁽³⁾ I/O 端口	TBD	—	V	TBD, $V_{DD} = 3.3\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D090B			—	—		
D090C			—	—		
D092		OSC2/CLKO (EC 模式)	$V_{DD} - 0.7$	—	V	$I_{OH} = -1.3\text{ mA}$, $V_{DD} = 3.3\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D100 ⁽⁴⁾	COSC2	输出引脚上的 容性负载规范 OSC2 引脚	—	15	pF	适用于使用外部时钟 驱动 OSC1 的 HS 模式
D101	CIO	所有的 I/O 引脚和 OSC2 引脚	—	50	pF	满足交流时序规范
D102	CB	SCLx 和 SDAx	—	400	pF	I ² C™ 规范

图注: TBD = 待定

- 注 1: 在 RC 振荡器配置中, OSC1/CLKI 引脚被配置为施密特触发器输入。在 RC 模式中建议不要使用外部时钟驱动 PICmicro® 器件。
- 注 2: MCLR 引脚上的泄漏电流主要取决于施加的电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。
- 注 3: 负电流定义为引脚输出的电流。

表 23-1: 存储器编程要求

直流规范			标准工作条件（除非另行声明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级）				
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
闪存程序存储器							
D130	EP	耐擦写能力	100	1K	—	E/W	-40°C 至 $+85^{\circ}\text{C}$
D131	VPR	用于读操作的 VDD	V _{MIN}	—	3.6	V	V _{MIN} = 最小工作电压
D132B	VPEW	用于自定时写操作的 VDD	V _{MIN}	—	3.6	V	V _{MIN} = 最小工作电压
D133A	T _{iw}	自定时写周期时间	—	10	—	ms	
D134	T _{RETD}	保存时间	10	20	—	年	假如没有违反其他规范
D135	IDDP	编程期间的供电电流	—	10	—	mA	

† 除非另行声明，否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。这些参数仅作为设计参考，未经测试。

PIC18F45J10 系列

表 23-2: 比较器规范

工作条件: $3.0V < V_{DD} < 3.6V$, $-40^{\circ}C < T_A < +85^{\circ}C$ (除非另行声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D300	V _{IOFF}	输入失调电压	—	±5.0	±10	mV	
D301	V _{ICM}	输入共模电压 *	0	—	$V_{DD} - 1.5$	V	
D302	CMRR	共模抑制比 *	55	—	—	dB	
300	T _{RESP}	响应时间 (1)*	—	150	400	ns	
301	T _{MC2OV}	比较器模式改变到输出有效之间的时间 *	—	—	10	μs	

* 这些参数仅为特性值, 未经测试。

注 1: 响应时间是在比较器一个输入端的电压为 $(V_{DD} - 1.5)/2$, 而另一个输入端的电压从 V_{SS} 变化到 V_{DD} 时测得的。

表 23-3: 参考电压规范

工作条件: $-40^{\circ}C < T_A < +85^{\circ}C$ (除非另行声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D310	V _{RES}	分辨率	$V_{DD}/24$	—	$V_{DD}/32$	LSb	
D311	V _{RAA}	绝对精度	—	—	1/2	LSb	
D312	V _{RUR}	单位电阻值 (R)	—	2k	—	Ω	
310	T _{SET}	稳定时间 (1)	—	—	10	μs	

注 1: 稳定时间是在 $CVRR = 1$ 并且 $CVR3:CVR0$ 从 “0000” 转变到 “1111” 时测得的。

表 23-4: 内部稳压器规范

工作条件: $-40^{\circ}C < T_A < +85^{\circ}C$ (除非另行声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	V _{RGOUT}	稳压器输出电压	—	2.5	—	V	
	CEFC	外部滤波器电容值	1	10	—	μF	必须是低 ESR 的电容

* 这些参数仅为特性值, 未经测试。尚未给这些规范分配参数编号。

23.4 AC (时序) 规范

23.4.1 时序参数符号

可根据以下任一格式来创建时序参数符号:

- | | | |
|-------------|-----------|---------------------------|
| 1. TppS2ppS | 3. Tcc:ST | (仅用于 I ² C 规范) |
| 2. TppS | 4. Ts | (仅用于 I ² C 规范) |

T		
F	频率	T 时间

小写字母 (pp) 及其含意:

pp		
cc	CCP1	osc OSC1
ck	CLKO	rd \overline{RD}
cs	\overline{CS}	rw \overline{RD} 或 \overline{WR}
di	SDI	sc SCK
do	SDO	ss \overline{SS}
dt	数据输入	t0 T0CKI
io	I/O 端口	t1 T1CKI
mc	MCLR	wr \overline{WR}

大写字母及其含意:

S		
F	下降	P 周期
H	高	R 上升
I	无效 (高阻态)	V 有效
L	低	Z 高阻态
仅用于 I ² C 模式		
AA	输出访问	High 高
BUF	总线空闲	Low 低

Tcc: ST (仅用于 I²C 规范)

CC		
HD	保持	SU 建立
ST		
DAT	数据输入保持	STO 停止条件
STA	启动条件	

PIC18F45J10 系列

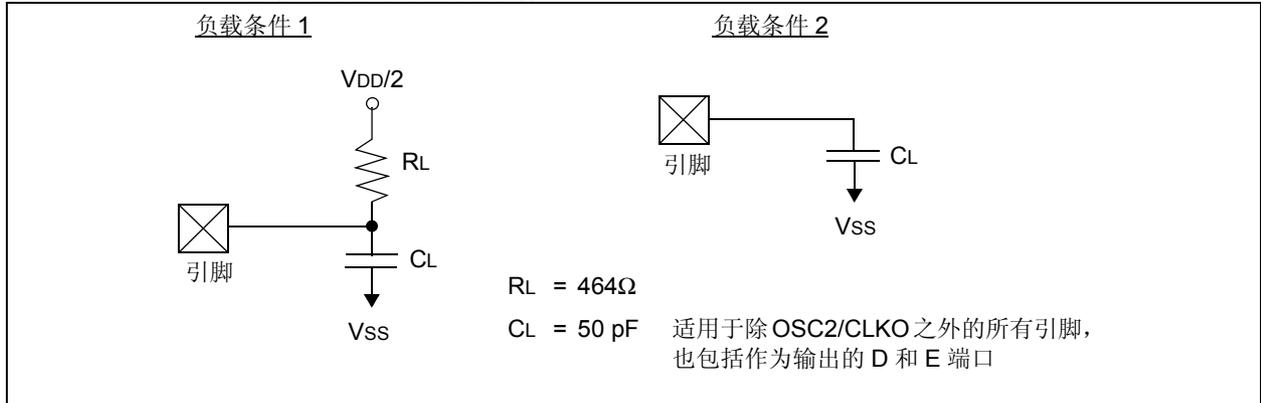
23.4.2 时序条件

除非另行声明，表 23-5 中指定的温度和电压适用于所有时序规范。图 23-3 规定了时序规范的负载条件。

表 23-5: 温度和电压规范——交流

交流规范	标准工作条件（除非另行声明）
	工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级）
	直流规范第 23.1 节和第 23.3 节说明了工作电压 V_{DD} 的范围。

图 23-3: 器件时序规范的负载条件



23.4.3 时序图和规范

图 23-4: 外部时钟时序 (除了 PLL 之外的所有模式)

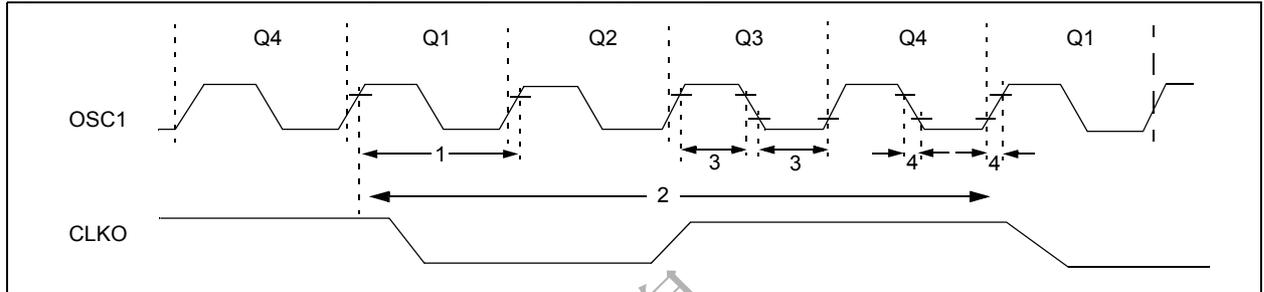


表 23-6: 外部时钟时序要求

参数编号	符号	特性	最小值	最大值	单位	条件
1A	FOSC	外部 CLKI 频率 (1)	DC	40	MHz	HS 振荡器模式
		振荡器频率 (1)	DC	40	MHz	HS 振荡器模式
1	Tosc	外部 CLKI 周期 (1)	25	—	ns	HS 振荡器模式
		振荡器周期 (1)	25	250	ns	HS 振荡器模式
2	Tcy	指令周期时间 (1)	100	—	ns	Tcy = 4/FOSC, 工业级
3	TosL, TosH	外部时钟输入 (OSC1) 高电平或低电平时间	10	—	ns	HS 振荡器模式
4	TosR, TosF	外部时钟输入 (OSC1) 上升或下降时间	—	7.5	ns	HS 振荡器模式

注 1: 对于所有配置, 指令周期时间 (Tcy) 等于输入振荡器时钟周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码所对应的特定振荡器类型的特征数据。超过这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期。所有器件在测试“最小值”时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC”(没有时钟)。

PIC18F45J10 系列

表 23-7: PLL 时钟时序规范 (VDD=2.5V 至 3.6V)

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
F10	FOSC	振荡器频率范围	4	—	10	MHz	仅 HS 模式
F11	Fsys	片上 VCO 系统频率	16	—	40	MHz	仅 HS 模式
F13	ΔCLK	CLKO 稳定性 (抗抖动)	-2	—	+2	%	

† 除非另行声明, 否则“典型值”栏中的数据均为 5.0V, 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

表 23-8: 交流规范: 内部 RC 精度
PIC18F24J10/25J10/44J10/45J10 (工业级)

参数编号	特性	最小值	典型值	最大值	单位	条件
	频率为 31 kHz 时的 INTRC 精度 ⁽¹⁾	26.562	—	35.938	kHz	-40°C 到 +85°C, VDD = 2.0 – 3.3V

注 1: 校准后的 INTRC 频率。INTRC 频率随 VDD 的改变而改变。

图 23-5: CLKO 和 I/O 时序

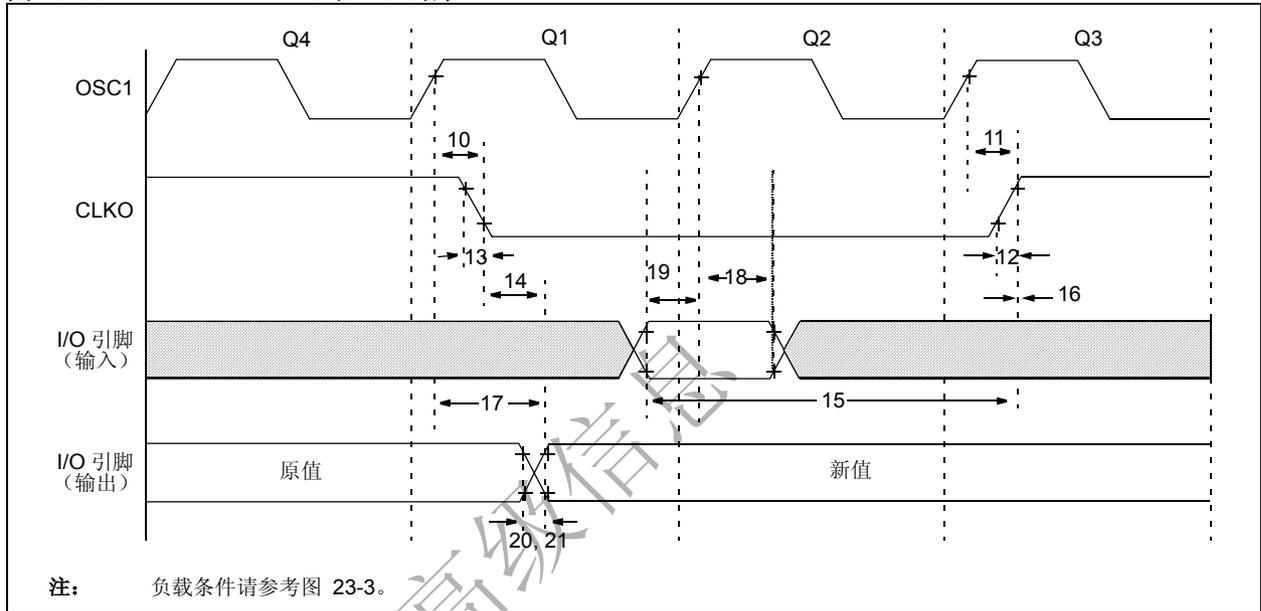


表 23-9: CLKO 和 I/O 时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
10	TosH2ckL	OSC1 ↑ 到 CLKO ↓ 的时间	—	TBD	TBD	ns	
11	TosH2ckH	OSC1 ↑ 到 CLKO ↑ 的时间	—	TBD	TBD	ns	
12	TckR	CLKO 上升时间	—	TBD	TBD	ns	
13	TckF	CLKO 下降时间	—	TBD	TBD	ns	
14	TckL2ioV	CLKO ↓ 至端口输出有效的时间	—	—	0.5 T _{CY} + 20	ns	
15	TioV2ckH	在出现 CLKO ↑ 前端口输入有效的的时间	0.25 T _{CY} + 25	—	—	ns	
16	TckH2ioI	在出现 CLKO ↑ 后端口输入数据的保持时间	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 周期) 至端口输出有效的的时间	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 周期) 至端口输入无效的时间 (I/O 输入保持时间)	100	—	—	ns	
18A			200	—	—	ns	
19	TioV2osH	端口输入有效到出现 OSC1 ↑ 的时间 (I/O 输入建立时间)	0	—	—	ns	
20	TioR	端口输出上升时间	—	TBD	TBD	ns	
20A			—	—	TBD	ns	
21	TioF	端口输出下降时间	—	TBD	TBD	ns	
21A			—	—	TBD	ns	
22†	TINP	INT 引脚高电平或低电平时间	T _{CY}	—	—	ns	
23†	TRBP	RB7:RB4 引脚电平变化中断信号的高电平或低电平时间	T _{CY}	—	—	ns	

图注： TBD = 待定

† 这些参数是与任何内部时钟边沿无关的异步事件。

PIC18F45J10 系列

图 23-6: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

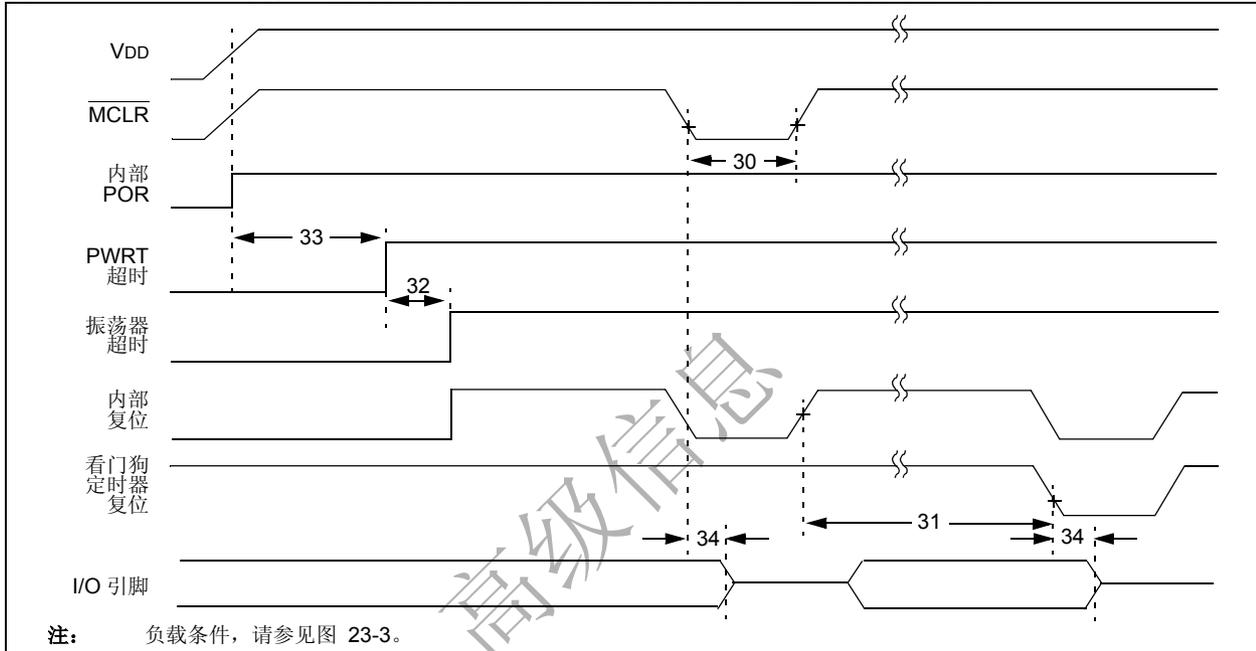


图 23-7: 欠压复位时序

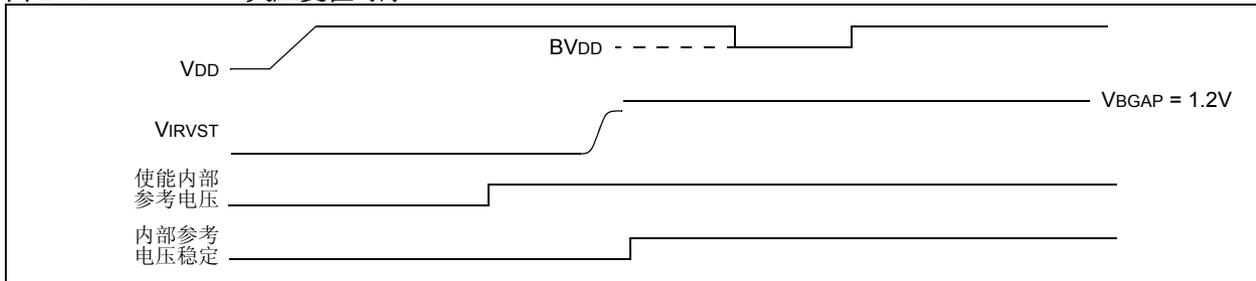


表 23-10: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
30	TMCL	MCLR 脉冲宽度 (低电平)	2	—	—	μs	
31	TWDT	看门狗定时器周期 (无后分频器)	TBD	TBD	TBD	ms	
32	TOST	振荡器起振定时器周期	1024 T _{osc}	—	1024 T _{osc}	—	T _{osc} = OSC1 周期
33	TPWRT	上电延时定时器周期	55.6	65.5	75	ms	
34	TIOZ	MCLR 低电平或看门狗定时器复位引起的 I/O 高阻态	—	2	—	μs	
38	TCSD	CPU 的启动时间	—	200	—	μs	
39	TIOBST	INTRC 电路稳定时间	—	TBD	—	μs	

图注: TBD = 待定

图 23-8: **TIMER0 和 TIMER1 外部时钟时序**

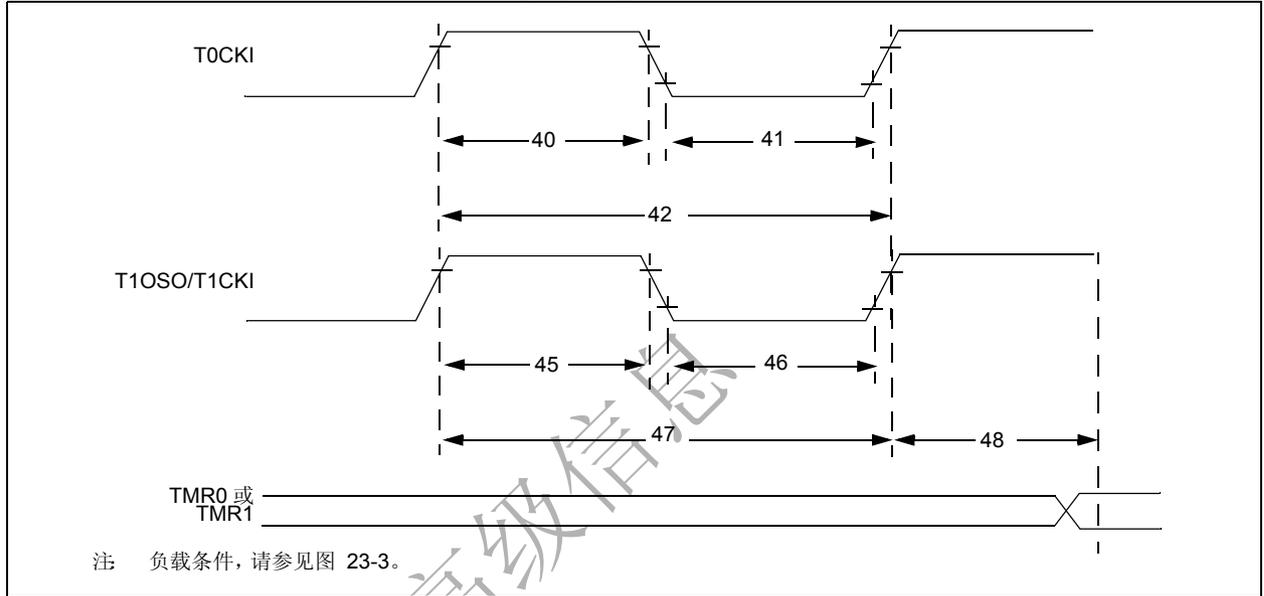


表 23-11: **TIMER0 和 TIMER1 外部时钟要求**

参数编号	符号	特性		最小值	最大值	单位	条件
40	Tt0H	TOCKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			有预分频器	10	—	ns	
41	Tt0L	TOCKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			有预分频器	10	—	ns	
42	Tt0P	TOCKI 周期	无预分频器	$T_{CY} + 10$	—	ns	
			有预分频器	取以下两者中的较大值: 20 ns 或 $(T_{CY} + 40) / N$	—	ns	
45	Tt1H	T1CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	ns	
			同步, 有预分频器	10	—	ns	
			异步	30	—	ns	
46	Tt1L	T1CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 5$	—	ns	
			同步, 有预分频器	10	—	ns	
			异步	30	—	ns	
47	Tt1P	T1CKI 输入周期	同步	取以下两者中的较大值: 20 ns 或 $(T_{CY} + 40) / N$	—	ns	N = 预分频值 (1, 2, 4, 8)
			异步	60	—	ns	
	Ft1	T1CKI 振荡器输入频率范围		DC	50	kHz	
48	TCKE2TMRI	外部 T1CKI 时钟边沿到定时器递增的延时		$2 T_{OSC}$	$7 T_{OSC}$	—	

PIC18F45J10 系列

图 23-9: 捕捉 / 比较 / PWM 时序 (包括 ECCP 模块)

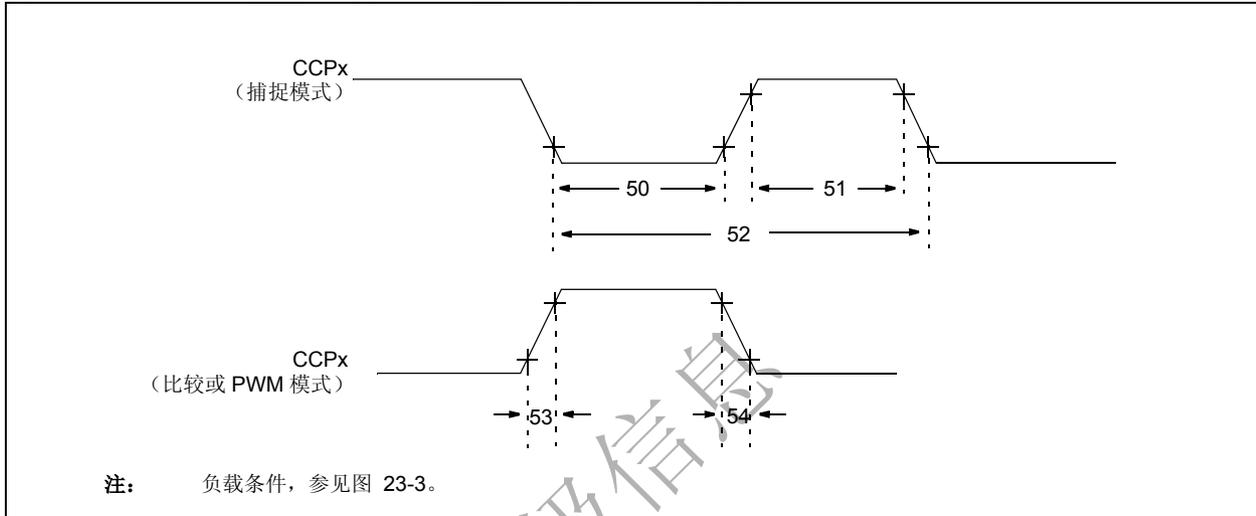


表 23-12: 捕捉 / 比较 / PWM 要求 (包括 ECCP 模块)

参数编号	符号	特性	最小值	最大值	单位	条件	
50	TccL	CCPx 输入低电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns	
		有预分频器	10	—	ns		
51	TccH	CCPx 输入高电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns	
		有预分频器	10	—	ns		
52	TccP	CCPx 输入周期	$\frac{3 T_{CY} + 40}{N}$	—	ns	N= 预分频值 (1、4 或 16)	
53	TccR	CCPx 输出上升时间	—	25	ns		
54	TccF	CCPx 输出下降时间	—	25	ns		

表 23-13: 并行从动端口要求

参数编号	符号	特性	最小值	最大值	单位	条件
62	TdtV2wrH	$\overline{WR} \uparrow$ 或 $\overline{CS} \uparrow$ 之前的数据输入有效时间 (建立时间)	20	—	ns	
63	TwrH2dtl	$\overline{WR} \uparrow$ 或 $\overline{CS} \uparrow$ 至数据输入无效的时间 (保持时间)	20	—	ns	
64	TrdL2dtV	$\overline{RD} \downarrow$ 和 $\overline{CS} \downarrow$ 至数据输出有效的时间	—	80	ns	
65	TrdH2dtl	$\overline{RD} \downarrow$ 或 $\overline{CS} \downarrow$ 至数据输出无效的时间	10	30	ns	
66	TibfINH	禁止 IBF 标志位被 $\overline{WR} \uparrow$ 或 $\overline{CS} \uparrow$ 清零的时间	—	$3 T_{CY}$		

图 23-10: SPI 主控模式时序示例 (CKE = 0)

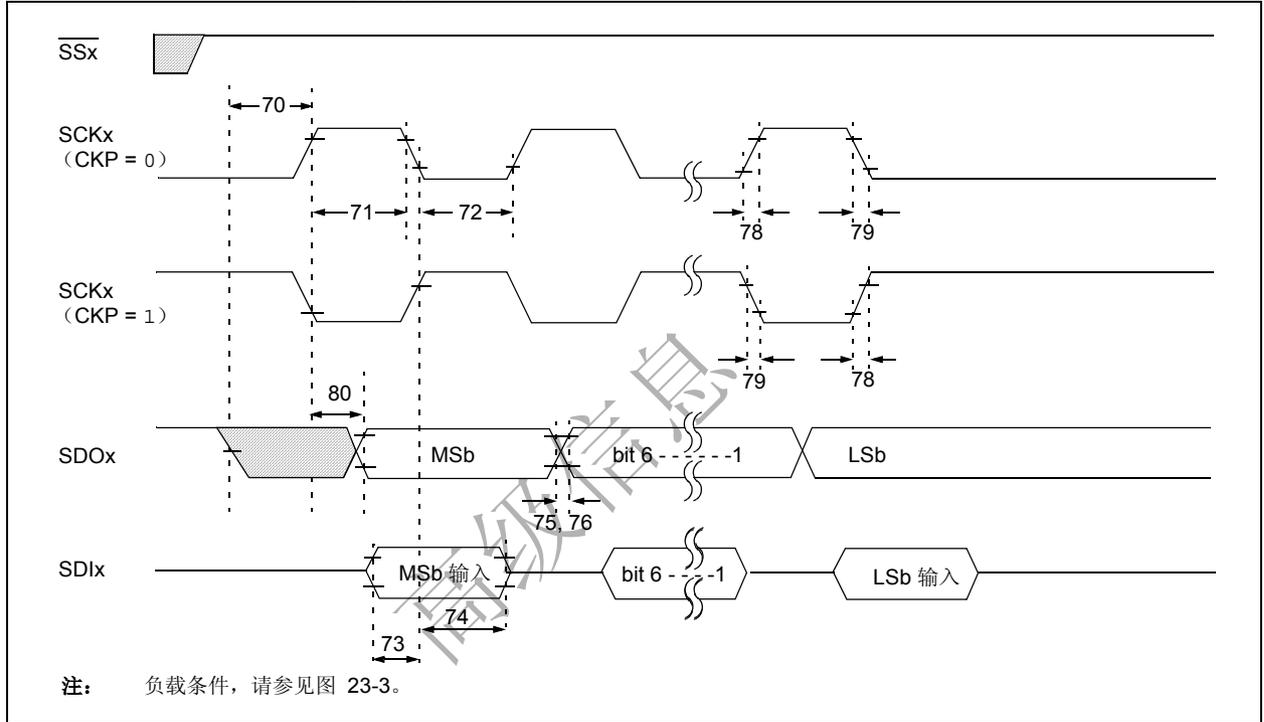


表 23-14: SPI 模式要求示例 (主控模式, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2sch, TssL2scl	SSx ↓ 到出现 SCKx ↓ 或 SCKx ↑ 输入的时间	Tcy	—	ns	
71	Tsch	SCKx 输入高电平时间 (从动模式)	连续	1.25 Tcy + 30	—	ns
71A			单字节	40	—	ns (注 1)
72	TscL	SCKx 输入低电平时间 (从动模式)	连续	1.25 Tcy + 30	—	ns
72A			单字节	40	—	ns (注 1)
73	TdiV2sch, TdiV2scl	SDIx 数据输入到 SCKx 边沿的建立时间	100	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	100	—	ns	
75	TdoR	SDOx 数据输出上升时间	—	25	ns	
76	TdoF	SDOx 数据输出下降时间	—	25	ns	
78	TscR	SCKx 输出上升时间 (主控模式)	—	25	ns	
79	TscF	SCKx 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, TscL2doV	在出现 SCKx 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	

注 1: 要求使用参数 #73A。

注 2: 仅当使用参数 #71A 和 #72A 时。

PIC18F45J10 系列

图 23-11: SPI 主控模式时序示例 (CKE = 1)

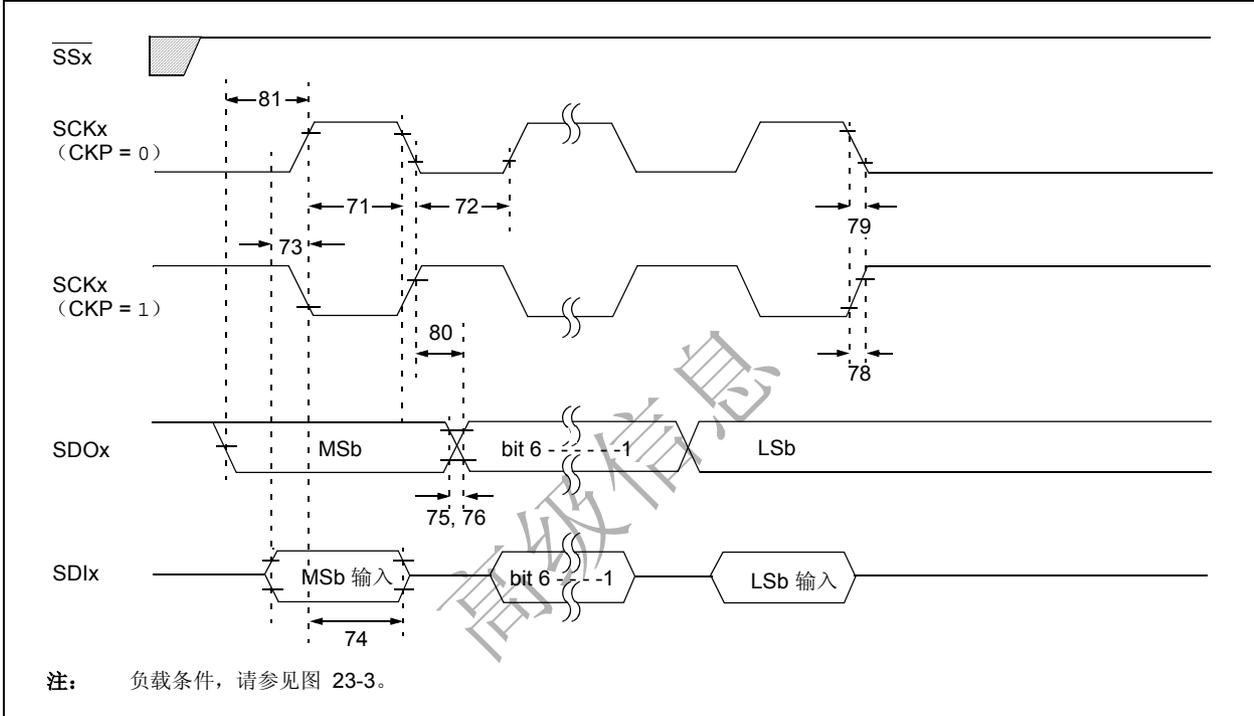


表 23-15: SPI 模式要求示例 (主控模式, CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
71	Tsch	SCKx 输入高电平时间	1.25 Tcy + 30	—	ns	
71A		(从动模式) 连续 单字节	40	—	ns	(注 1)
72	Tscl	SCKx 输入低电平时间	1.25 Tcy + 30	—	ns	
72A		(从动模式) 连续 单字节	40	—	ns	(注 1)
73	TdIV2sch, TdIV2scl	SDIx 数据输入到 SCKx 边沿的建立时间	100	—	ns	
73A	Tb2b	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, Tscl2diL	SDIx 数据输入到 SCKx 边沿的保持时间	100	—	ns	
75	TdoR	SDOx 数据输出上升时间	—	25	ns	
76	TdoF	SDOx 数据输出下降时间	—	25	ns	
78	TscR	SCKx 输出上升时间 (主控模式)	—	25	ns	
79	TscF	SCKx 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, Tscl2doV	在出现 SCKx 边沿之后 SDOx 数据输出有效的时间	—	50	ns	
81	TdoV2sch, TdoV2scl	SDOx 输出数据建立完成到出现 SCKx 边沿的时间	Tcy	—	ns	

注 1: 要求使用参数 #73A。

注 2: 仅当使用参数 #71A 和 #72A 时。

图 23-12: SPI 从动模式时序示例 (CKE = 0)

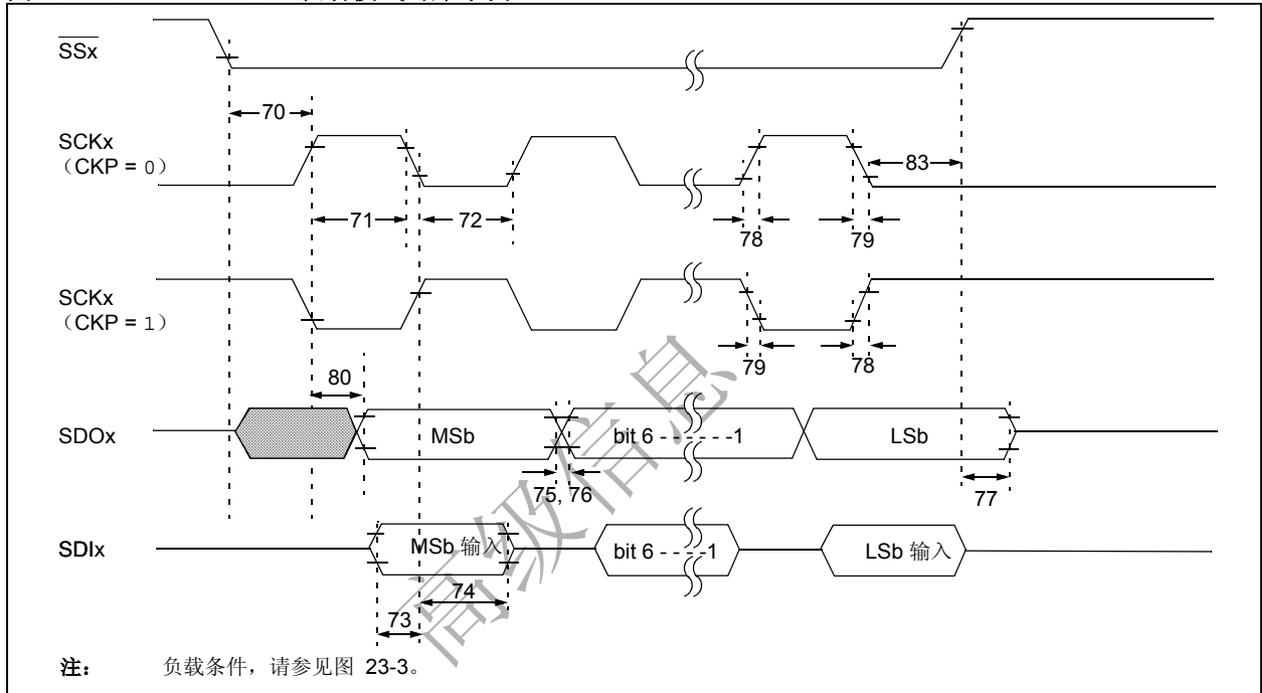


表 23-16: SPI 模式要求示例 (从动模式时序, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	从出现 $\overline{SSx} \downarrow$ 到出现 SCKx \downarrow 或 SCKx \uparrow 输入的时间	T _{CY}	—	ns	
71	Tsch	SCKx 输入高电平时间 (从动模式)	连续	1.25 T _{CY} + 30	—	ns
71A			单字节	40	—	ns (注 1)
72	TscL	SCKx 输入低电平时间 (从动模式)	连续	1.25 T _{CY} + 30	—	ns
72A			单字节	40	—	ns (注 1)
73	TdiV2scH, TdiV2scL	SDIx 数据输入到出现 SCKx 边沿的建立时间	100	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	1.5 T _{CY} + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDIx 数据输入到出现 SCKx 边沿的保持时间	100	—	ns	
75	TdoR	SDOx 数据输出上升时间	—	25	ns	
76	TdoF	SDOx 数据输出下降时间	—	25	ns	
77	TssH2doZ	从出现 $\overline{SSx} \uparrow$ 到 SDOx 输出呈现高阻态的时间	10	50	ns	
78	TscR	SCKx 输出上升时间 (主控模式)	—	25	ns	
79	TscF	SCKx 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, TscL2doV	在 SCKx 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	
83	Tsch2ssH, TscL2ssH	在 SCKx 边沿后出现 $\overline{SSx} \uparrow$ 的时间	1.5 T _{CY} + 40	—	ns	

注 1: 要求使用参数 #73A。

注 2: 仅当使用参数 #71A 和 #72A 时。

PIC18F45J10 系列

图 23-13: SPI 从动模式时序示例 (CKE = 1)

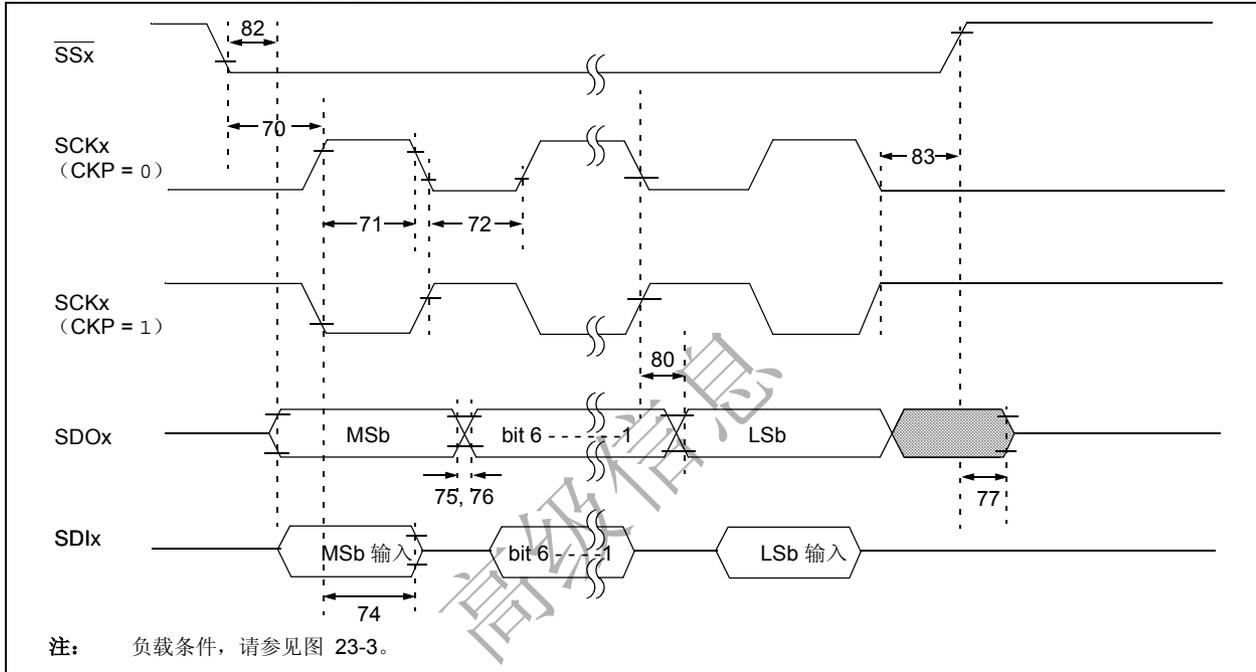


表 23-17: SPI 从动模式要求示例 (CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2sCH, TssL2sCL	从出现 $\overline{SSx} \downarrow$ 到出现 SCKx \downarrow 或 SCKx \uparrow 输入的时间	T_{CY}	—	ns	
71	Tsch	SCKx 输入高电平时间 (从动模式)	连续	$1.25 T_{CY} + 30$	—	ns
71A			单字节	40	—	ns (注 1)
72	TscL	SCKx 输入低电平时间 (从动模式)	连续	$1.25 T_{CY} + 30$	—	ns
72A			单字节	40	—	ns (注 1)
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	$1.5 T_{CY} + 40$	—	ns	(注 2)
74	Tsch2dIL, TscL2dIL	SDIx 数据输入到出现 SCKx 边沿的保持时间	100	—	ns	
75	TdoR	SDOx 数据输出上升时间	—	25	ns	
76	TdoF	SDOx 数据输出下降时间	—	25	ns	
77	TssH2doZ	从出现 $\overline{SSx} \uparrow$ 到 SDOx 输出呈现高阻态的时间	10	50	ns	
78	TscR	SCKx 输出上升时间 (主控模式)	—	25	ns	
79	TscF	SCKx 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, TscL2doV	在出现 SCKx 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	
82	TssL2doV	在出现 $\overline{SSx} \downarrow$ 之后 SDOx 数据输出有效的的时间	—	50	ns	
83	Tsch2ssH, TscL2ssH	在 SCKx 边沿后出现 $\overline{SSx} \uparrow$ 的时间	$1.5 T_{CY} + 40$	—	ns	

注 1: 要求使用参数 #73A。

注 2: 仅当使用参数 #71A 和 #72A 时。

图 23-14: I²C™ 总线启动 / 停止位时序

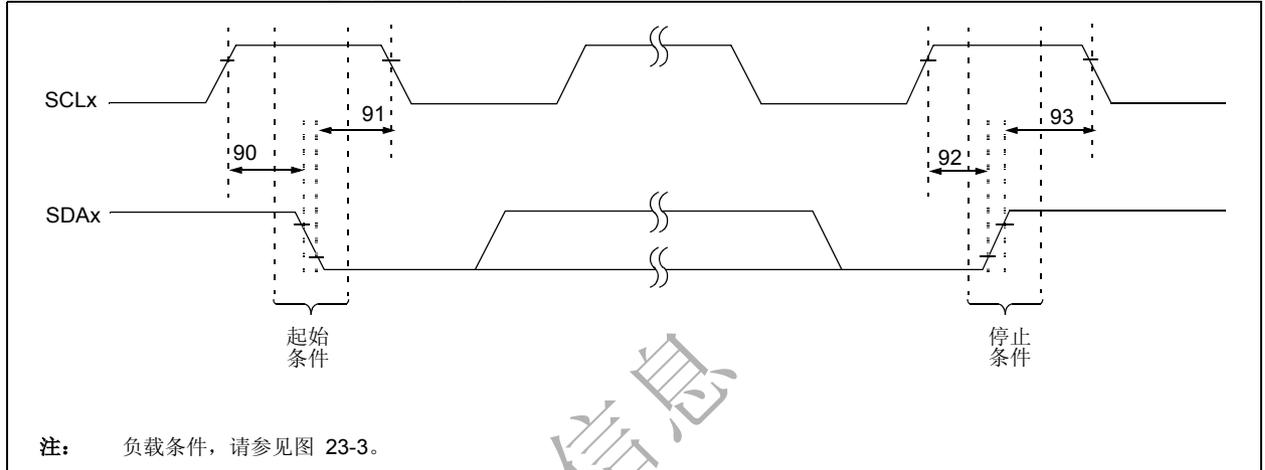
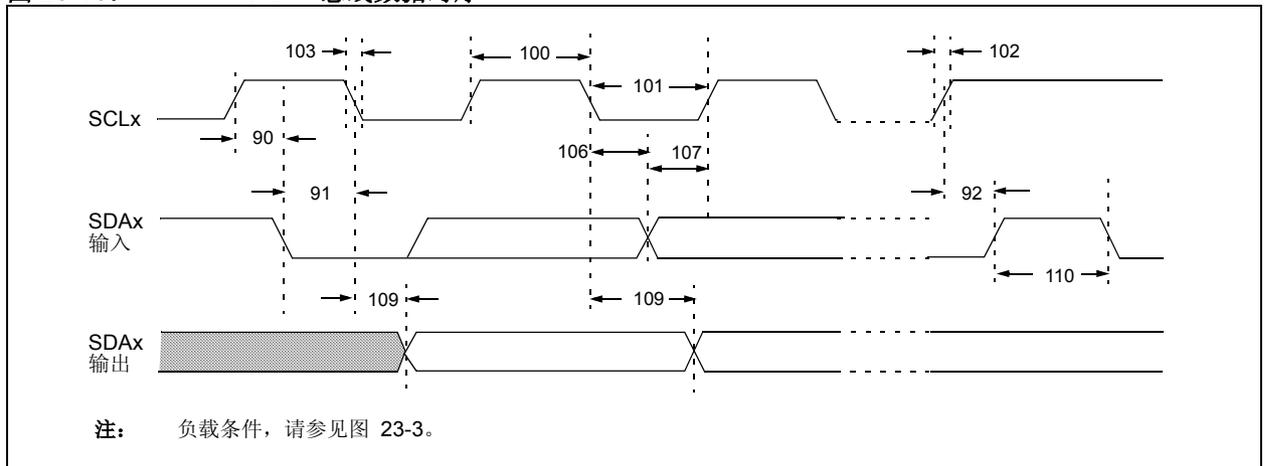


表 23-18: I²C™ 总线启动 / 停止位要求 (从动模式)

参数编号	符号	特性	最小值	最大值	单位	条件	
90	TSU:STA	启动条件建立时间	100 kHz 模式	4700	—	ns	仅与重复启动条件相关
			400 kHz 模式	600	—		
91	THD:STA	启动条件保持时间	100 kHz 模式	4000	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	600	—		
92	TSU:STO	停止条件建立时间	100 kHz 模式	4700	—	ns	
			400 kHz 模式	600	—		
93	THD:STO	停止条件保持时间	100 kHz 模式	4000	—	ns	
			400 kHz 模式	600	—		

图 23-15: I²C™ 总线数据时序



PIC18F45J10 系列

表 23-19: I²C™ 总线数据要求 (从动模式)

参数编号	符号	特性	最小值	最大值	单位	条件	
100	T _{HIGH}	时钟高电平时间	100 kHz 模式	4.0	—	μs	PIC18F45J10 的工作频率不得低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	PIC18F45J10 的工作频率不得低于 10 MHz
			MSSP 模块	1.5 T _{CY}	—		
101	T _{LOW}	时钟低电平时间	100 kHz 模式	4.7	—	μs	PIC18F45J10 的工作频率不得低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	PIC18F45J10 的工作频率不得低于 10 MHz
			MSSP 模块	1.5 T _{CY}	—		
102	T _R	SDA _x 和 SCL _x 的上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1 C _B	300	ns	C _B 值被规定在 10 到 400 pF 之间
103	T _F	SDA _x 和 SCL _x 的下降时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	20 + 0.1 C _B	300	ns	C _B 值被规定在 10 到 400 pF 之间
90	T _{SU:STA}	启动条件建立时间	100 kHz 模式	4.7	—	μs	仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs	
91	T _{HD:STA}	启动条件保持时间	100 kHz 模式	4.0	—	μs	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
106	T _{HD:DAT}	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
107	T _{SU:DAT}	数据输入建立时间	100 kHz 模式	250	—	ns	(注 2)
			400 kHz 模式	100	—	ns	
92	T _{SU:STO}	停止条件建立时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	0.6	—	μs	
109	T _{AA}	时钟输出有效时间	100 kHz 模式	—	3500	ns	(注 1)
			400 kHz 模式	—	—	ns	
110	T _{BUF}	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
D102	C _B	总线的容性负载	—	400	pF		

注 1: 为避免产生意外的启动或停止条件, 作为发送器的器件必须提供这个内部最小延时以覆盖 SCL 下降沿的未定义区域 (最小值 300ns)。

注 2: 快速模式的 I²C™ 总线器件也可在标准模式的 I²C 总线系统中使用, 但必须满足 T_{SU:DAT} ≥ 250 ns 的要求。如果快速模式器件没有延长 SCL_x 信号的低电平周期, 则必然满足此条件。如果该器件延长了 SCL_x 信号的低电平周期, 其下一个数据位必须输出到 SDA_x 线。SCL 线被释放前, 根据标准模式 I²C 总线规范, T_{R max.} + T_{SU:DAT} = 1000 + 250 = 1250 ns。

图 23-16: 主控 SSP I²C™ 总线启动 / 停止位时序波形

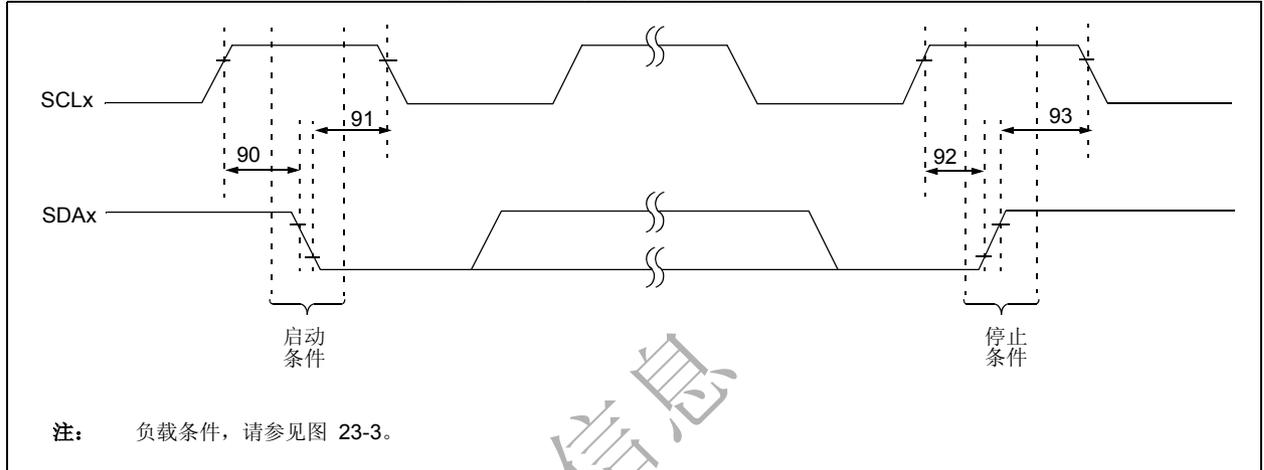
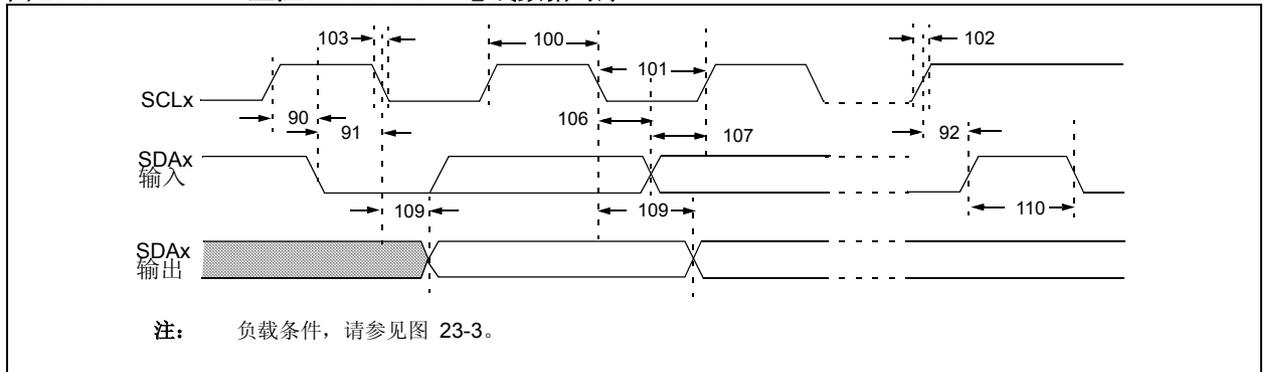


表 23-20: 主控 SSP I²C™ 总线启动 / 停止位要求

参数编号	符号	特性	最小值	最大值	单位	条件
90	TSU:STA	启动条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns 仅与重复启动条件有关
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—	
91	THD:STA	启动条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns 这个周期后产生第一个时钟脉冲
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—	
92	TSU:STO	停止条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—	
93	THD:STO	停止条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—	

注 1: 对于所有 I²C™ 引脚, 最小引脚电容均为 10 pF。

图 23-17: 主控 MSSP I²C™ 总线数据时序



PIC18F45J10 系列

表 23-21: 主控 SSP I²C™ 总线数据要求

参数编号	符号	特性	最小值	最大值	单位	条件
100	THIGH	时钟高电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
101	TLOW	时钟低电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
102	TR	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
			1 MHz 模式 (1)	—	300	ns
103	TF	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
			1 MHz 模式 (1)	—	100	ns
90	TSU:STA	启动条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
91	THD:STA	启动条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
106	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	ms
			1 MHz 模式 (1)	TBD	—	ns
107	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns
			400 kHz 模式	100	—	ns
			1 MHz 模式 (1)	TBD	—	ns
92	TSU:STO	停止条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
109	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns
			400 kHz 模式	—	1000	ns
			1 MHz 模式 (1)	—	—	ns
110	TBUF	总线空闲时间	100 kHz 模式	4.7	—	ms
			400 kHz 模式	1.3	—	ms
			1 MHz 模式 (1)	TBD	—	ms
D102	C _B	总线的容性负载	—	400	pF	

图注: TBD = 待定

注 1: 对于所有 I²C™ 引脚, 最小引脚电容均为 10 pF。

注 2: 快速模式的 I²C 总线器件可用于标准模式的 I²C 总线系统中, 但必须满足参数 #107 ≥ 250 ns 的要求。如果快速模式器件没有延长 SCLx 信号的低电平周期, 则必然满足此条件。如果该器件延长 SCLx 信号的低电平周期, 它必须将下一个数据位输出到 SDAx 线。SCLx 线被释放前, 在 100 kHz 模式下, 参数 #102 + 参数 #107 = 1000 + 250 = 1250 ns。

图 23-18: EUSART 同步发送 (主控 / 从动) 时序

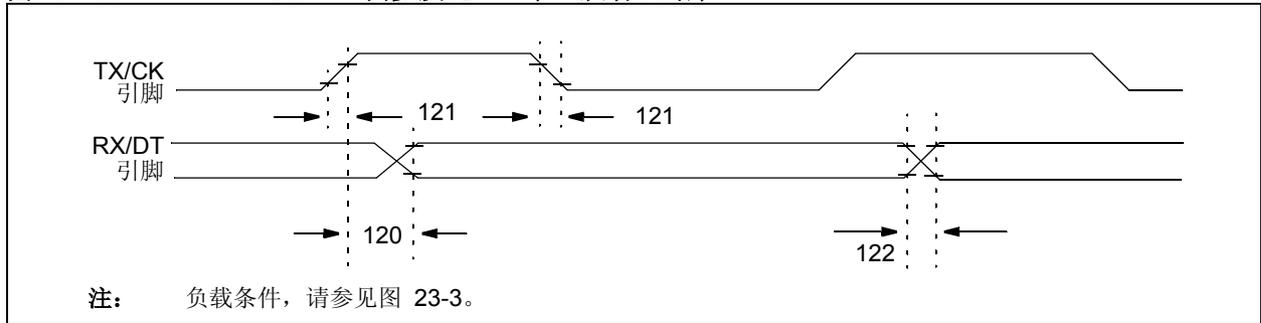


表 23-22: EUSART 同步发送要求

参数编号	符号	特性	最小值	最大值	单位	条件
120	TckH2DTV	同步发送 (主控和从动) 时钟高电平至数据输出有效的 时间	—	40	ns	
121	TCKRF	时钟输出上升时间和下降时间 (主控模式)	—	20	ns	
122	TDTRF	数据输出的上升时间和下降时间	—	20	ns	

图 23-19: EUSART 同步接收 (主控 / 从动) 时序

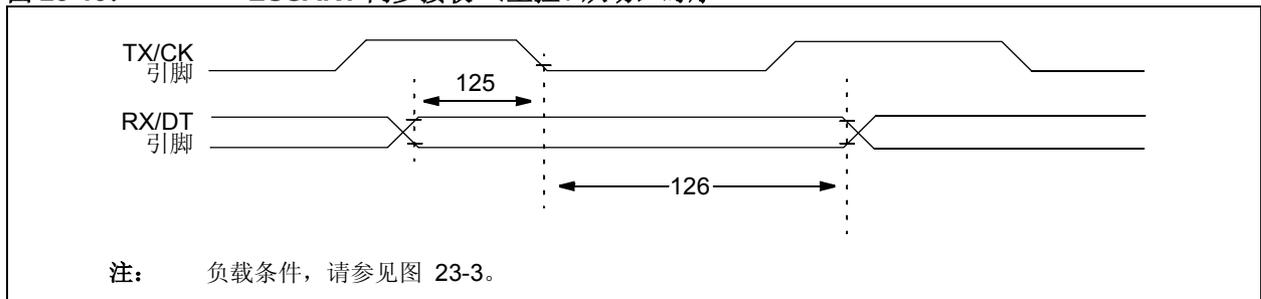


表 23-23: EUSART 同步接收要求

参数编号	符号	特性	最小值	最大值	单位	条件
125	TdtV2CKL	同步接收 (主控和从动) 在出现 CK ↓ 之前数据的保持时间 (DT 保持时间)	10	—	ns	
126	TckL2DTL	在出现 CK ↓ 之后数据的保持时间 (DT 保持时间)	15	—	ns	

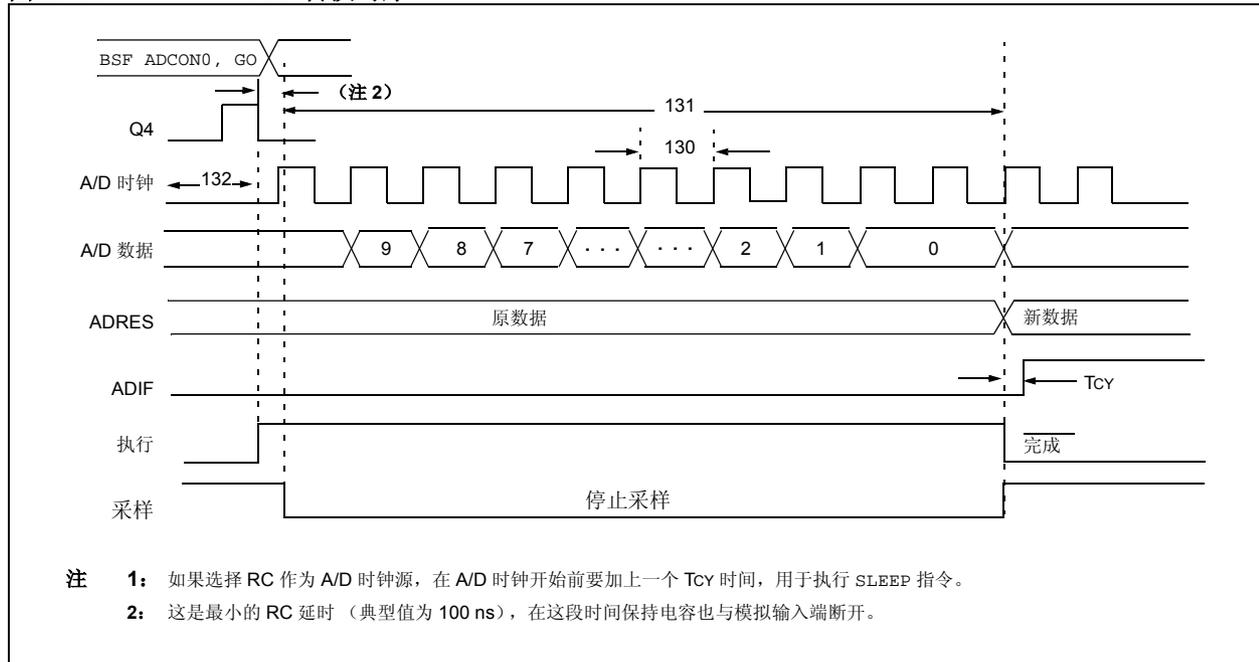
PIC18F45J10 系列

表 23-24: A/D 转换器规范: PIC18F24J10/25J10/44J10/45J10 (工业级)

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
A01	NR	分辨率	—	—	10	位	$\Delta V_{REF} \geq 3.0V$
A03	EIL	积分线性误差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	微分线性误差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A06	E0FF	失调误差	—	—	$< \pm 1.5$	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	增益误差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A10	—	单调性	保证 ⁽¹⁾			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	ΔV_{REF}	参考电压范围 ($V_{REFH} - V_{REFL}$)	1.8	—	—	V	$V_{DD} < 3.0V$
			3	—	—	V	$V_{DD} \geq 3.0V$
A21	V_{REFH}	参考电压高电平	V_{SS}	—	V_{REFH}	V	
A22	V_{REFL}	参考电压低电平	$V_{SS} - 0.3V$	—	$V_{DD} - 3.0V$	V	
A25	V_{AIN}	模拟输入电压	V_{REFL}	—	V_{REFH}	V	
A30	Z_{AIN}	建议的模拟电压源阻抗	—	—	2.5	k Ω	
A50	I_{REF}	V_{REF} 输入电流 ⁽²⁾	—	—	5	μA	在 V_{AIN} 采集期间。
			—	—	150	μA	在 A/D 转换期间。

- 注 1: A/D 转换结果不会因输入电压的递增而递减, 并且不会丢失编码。
 注 2: V_{REFH} 电流来自 RA3/AN3/ V_{REF+} 引脚或 VDD, 可以选择两者之一作为 V_{REFH} 源。
 V_{REFL} 电流来自 RA2/AN2/ V_{REF-} 引脚或 VSS, 可以选择两者之一作为 V_{REFL} 源。

图 23-20: A/D 转换时序



- 注 1: 如果选择 RC 作为 A/D 时钟源, 在 A/D 时钟开始前要加上一个 T_{cy} 时间, 用于执行 SLEEP 指令。
 注 2: 这是最小的 RC 延时 (典型值为 100 ns), 在这段时间保持电容也与模拟输入端断开。

表 23-25: A/D 转换要求

参数编号	符号	特性	最小值	最大值	单位	条件
130	TAD	A/D 时钟周期	0.7	25.0 ⁽¹⁾	μs	基于 TOSC, VREF ≥3.0V
			TBD	1	μs	A/D RC 模式
131	TCNV	转换时间 (不包括采集时间) (注 2)	11	12	TAD	
132	TACQ	采集时间 (注 3)	1.4	—	μs	-40° C 到 +85° C
			TBD	—	μs	0° C 到 +85° C
135	TSWC	从转换→采样的切换时间	—	(注 4)		

图注: TBD = 待定

- 注 1: A/D 时钟周期取决于器件频率和 TAD 时钟分频器。
- 注 2: ADRES 寄存器中的内容可在下一个 Tcy 周期读出。
- 注 3: 转换完成后当电压满量程变化时 (VDD 至 VSS, 或 VSS 至 VDD), 保持电容采集一个“新的”输入电压所需的时间。在输入通道上的源阻抗 (RS) 为 50Ω。
- 注 4: 在器件时钟的下一个周期。

PIC18F45J10 系列

注:

24.0 DC 和 AC 特性图表

当前没有可用图表。

PIC18F45J10 系列

注:

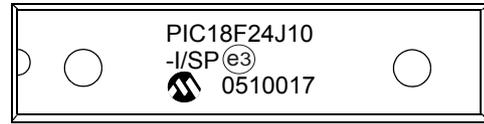
25.0 封装信息

25.1 封装标识信息

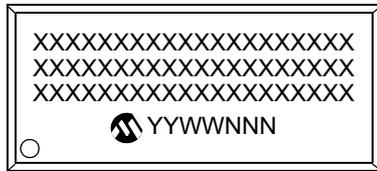
28 引脚 SPDIP



示例



28 引脚 SOIC



示例



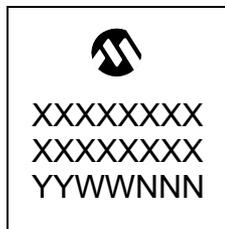
28 引脚 SSOP



示例



28 引脚 QFN



示例



图注:

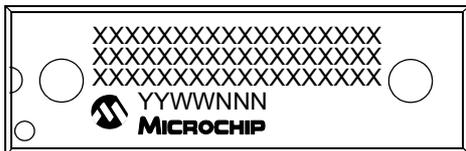
- XX...X 客户信息
- Y 年份代码（日历年的最后一位数字）
- YY 年份代码（日历年的最后两位数字）
- WW 星期代码（一月一日的星期代码为“01”）
- NNN 以字母数字排序的追踪代码
- (e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志
- * 表示无铅封装。JEDEC 无铅标志（(e3)）标示于此种封装的外包装上。

注: Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

PIC18F45J10 系列

封装标识信息 (续)

40 引脚 PDIP



示例



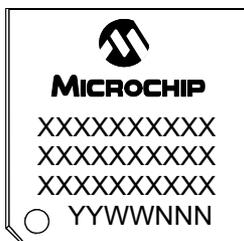
44 引脚 QFN



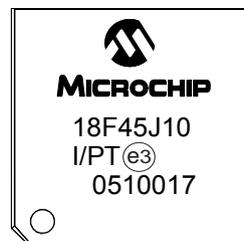
示例



44 引脚 TQFP



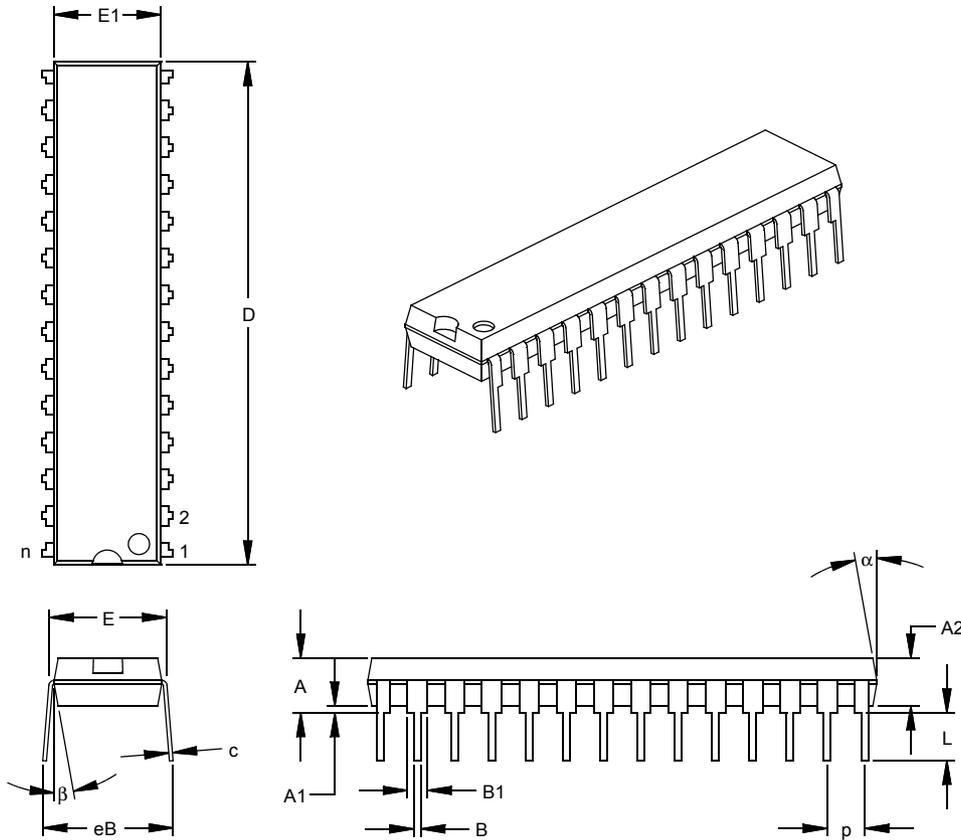
示例



25.2 封装详细信息

以下部分将介绍各种封装的技术细节。

28 引脚窄型塑封双列直插式封装 (SP) —— 主体 300 mil (PDIP)



尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n	28			28		
引脚间距	p		.100			2.54	
顶端到固定面高度	A	.140	.150	.160	3.56	3.81	4.06
塑模封装厚度	A2	.125	.130	.135	3.18	3.30	3.43
塑模底面到固定面高度	A1	.015			0.38		
肩到肩宽度	E	.300	.310	.325	7.62	7.87	8.26
塑模封装宽度	E1	.275	.285	.295	6.99	7.24	7.49
总长度	D	1.345	1.365	1.385	34.16	34.67	35.18
引脚尖到固定面高度	L	.125	.130	.135	3.18	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.040	.053	.065	1.02	1.33	1.65
引脚下部宽度	B	.016	.019	.022	0.41	0.48	0.56
总排列间距	§ eB	.320	.350	.430	8.13	8.89	10.92
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

§ 重要特性

注:

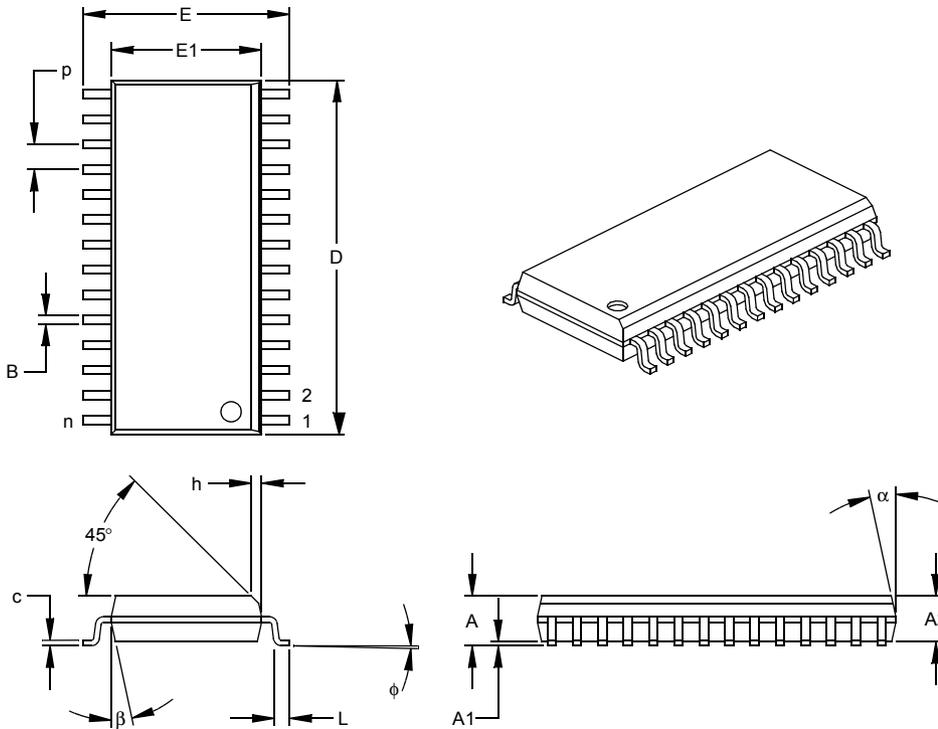
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254 毫米)。

等同于 JEDEC 号: MO-095

图号: C04-070

PIC18F45J10 系列

28 引脚宽型塑封小外形封装 (SO) —— 主体 300 mil (SOIC)

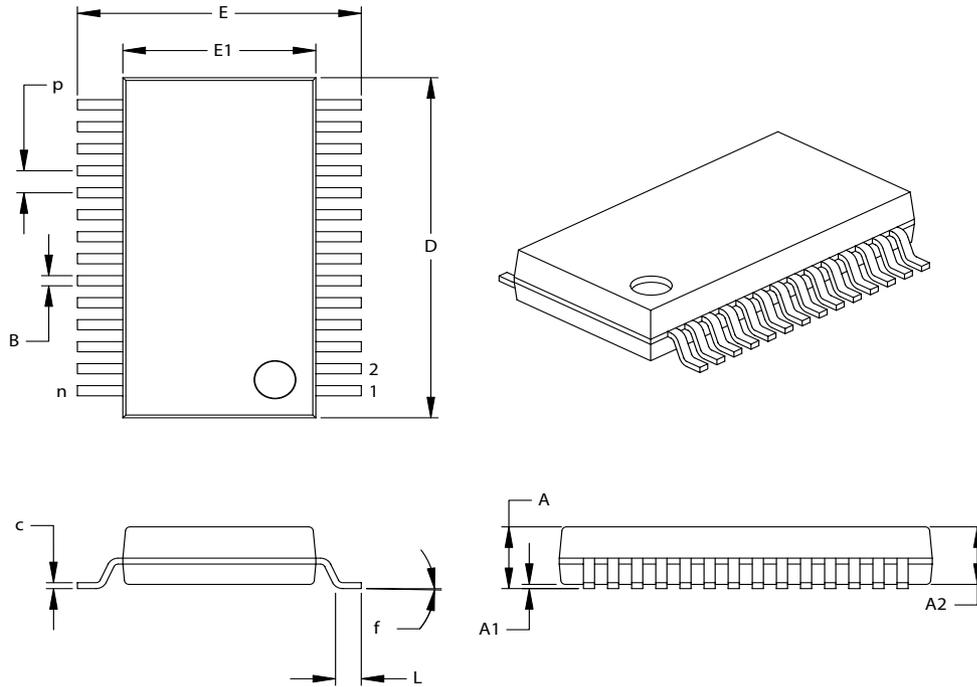


尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	p		.050			1.27	
总高度	A	.093	.099	.104	2.36	2.50	2.64
塑模封装厚度	A2	.088	.091	.094	2.24	2.31	2.39
悬空间隙 §	A1	.004	.008	.012	0.10	0.20	0.30
总宽度	E	.394	.407	.420	10.01	10.34	10.67
塑模封装宽度	E1	.288	.295	.299	7.32	7.49	7.59
总长度	D	.695	.704	.712	17.65	17.87	18.08
斜面投影距离	h	.010	.020	.029	0.25	0.50	0.74
底脚长度	L	.016	.033	.050	0.41	0.84	1.27
底脚倾角	φ	0	4	8	0	4	8
引脚厚度	c	.009	.011	.013	0.23	0.28	0.33
引脚宽度	B	.014	.017	.020	0.36	0.42	0.51
塑模顶部锥度	α	0	12	15	0	12	15
塑模底部锥度	β	0	12	15	0	12	15

* 控制参数
§ 重要特性

注：
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254 毫米)。
等同于 JEDEC 号: MS-013
图号: C04-052

28 引脚塑封缩小外形封装 (SS) —— 主体 209mil, 5.30 mm (SSOP)



尺寸范围	单位	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n	28			28		
引脚间距	p		.026			0.65	
总高度	A	-	-	.079	-	-	2.0
塑模封装厚度	A2	.065	.069	.073	1.65	1.75	1.85
悬空间隙	A1	.002	-	-	0.05	-	-
总宽度	E	.295	.307	.323	7.49	7.80	8.20
塑模封装厚度	E1	.009	.209	.220	5.00	5.30	5.60
总长度	D	.390	.402	.413	9.90	10.20	10.50
底角长度	L	.022	.030	.037	0.55	0.75	0.95
引脚厚度	c	.004	-	.010	0.09	-	0.25
底角倾斜角	f	0°	4°	8°	0°	4°	8°
引脚宽度	B	.009	-	.015	0.22	-	0.38

*控制参数

注:

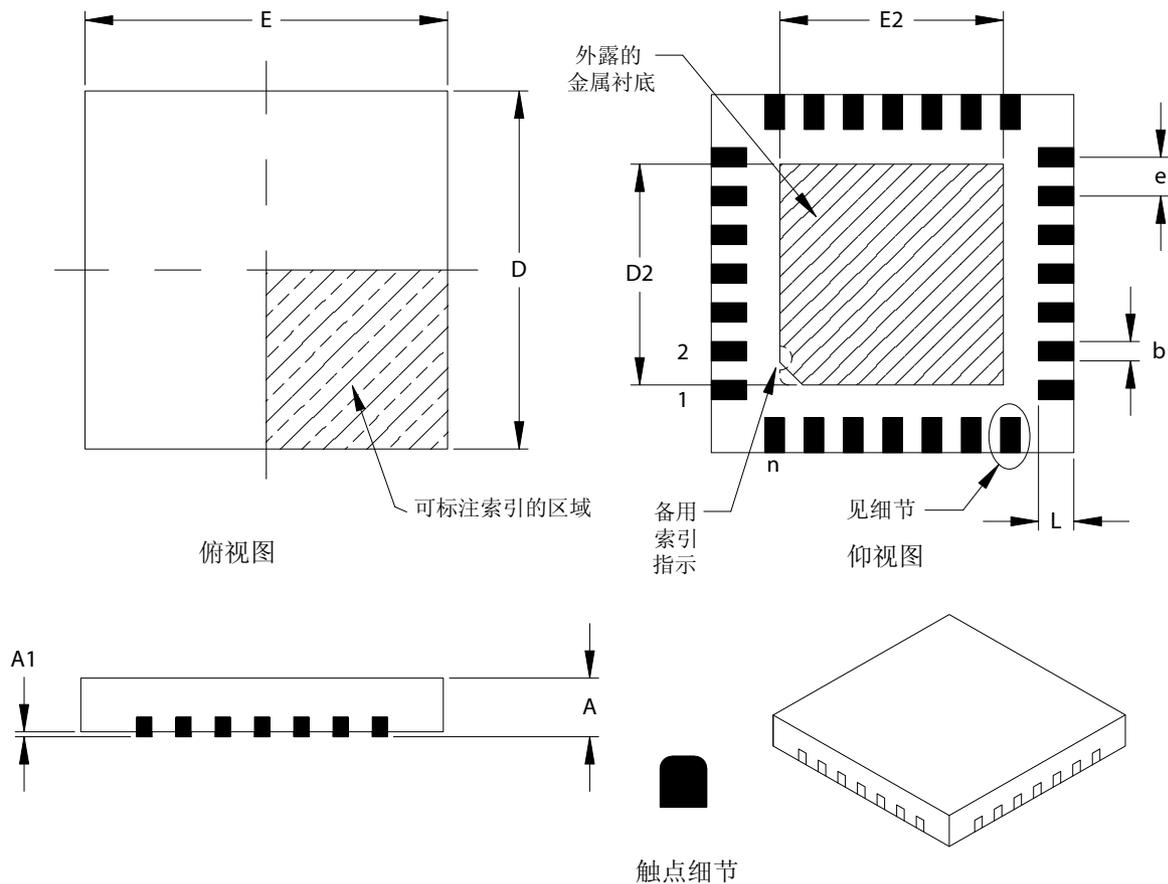
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254 毫米)。

等同于 JEDEC 号: MO-150

图号 C04-073

PIC18F45J10 系列

28 引脚塑封正方扁平无脚封装 (ML) 主体 6x6 mm (QFN) —— 触点长度为 0.55 mm (Saw Singulated)



尺寸范围	单位	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	e	.026 BSC			0.65 BSC		
总高度	A	.031	.035	.039	0.80	0.90	1.00
悬空间隙	A1	.000	.001	.002	0.00	0.02	0.05
引脚顶部到塑模底部距离	A3	.008 REF			0.20 REF		
总宽度	E	.232	.236	.240	5.90	6.00	6.10
外露金属衬底宽度	E2	.153	.167	.169	3.89	4.24	4.29
总长度	D	.232	.236	.240	5.90	6.00	6.10
外露金属衬底长度	D2	.153	.167	.169	3.89	4.24	4.29
触点宽度	b	.009	.011	.013	0.23	0.28	0.33
触点长度	L	.018	.022	.024	0.45	0.55	0.65

*控制参数

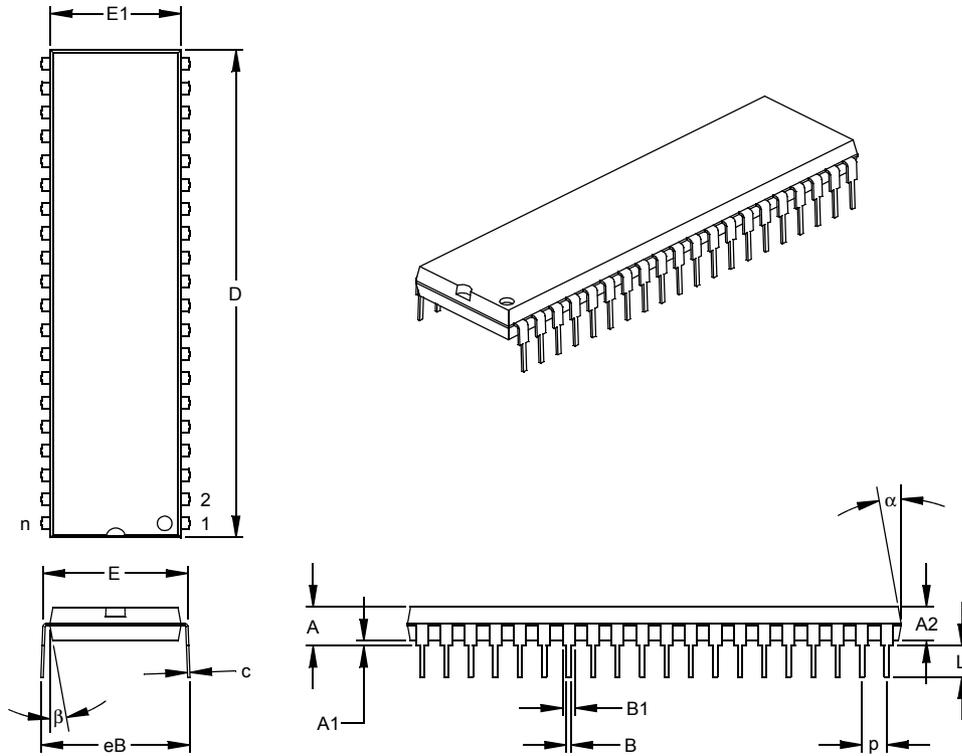
注:

外露金属衬底尺寸随管芯叶片大小而变化。

等同于JEDEC号: MO-220

图号: C04-105, 修订于 10-14-05

40 引脚塑封双列直插式封装 (P) —— 主体 600 mil (PDIP)



尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n		40			40	
引脚间距	p		.100			2.54	
顶端到固定面高度	A	.160	.175	.190	4.06	4.45	4.83
塑模封装厚度	A2	.140	.150	.160	3.56	3.81	4.06
塑模底面到固定面高度	A1	.015			0.38		
肩到肩宽度	E	.595	.600	.625	15.11	15.24	15.88
塑模封装宽度	E1	.530	.545	.560	13.46	13.84	14.22
总长度	D	2.045	2.058	2.065	51.94	52.26	52.45
引脚尖到固定面高度	L	.120	.130	.135	3.05	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.030	.050	.070	0.76	1.27	1.78
引脚下部宽度	B	.014	.018	.022	0.36	0.46	0.56
总排列间距 §	eB	.620	.650	.680	15.75	16.51	17.27
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

§ 重要特性

注:

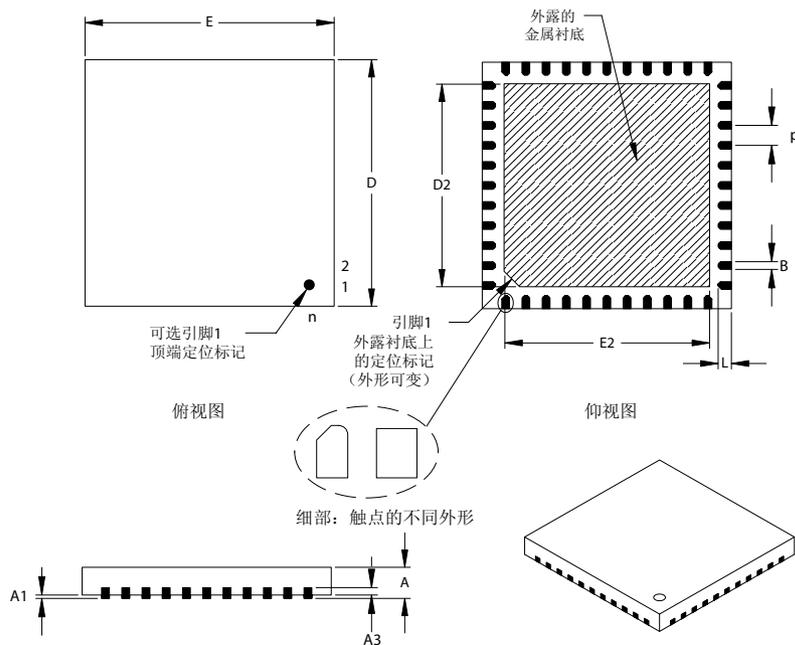
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254 毫米)。

等同于 JEDEC 号: MO-011

图号: C04-016

PIC18F45J10 系列

44 引脚塑封正方扁平无脚封装 (ML) —— 主体 8x8 mm (QFN)



尺寸范围	单位	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n		44			44	
引脚间距	P	.026 BSC ¹			0.65 BSC ¹		
总高度	A	.031	.035	.039	0.80	0.90	1.00
悬空间隙	A1	.000	.001	.002	0	0.02	0.05
引脚顶部到塑模底部距离	A3	.010 REF ²			0.25 REF ²		
总宽度	E	.309	.315	.321	7.85	8.00	8.15
外露金属衬底宽度	E2	.236	.258	.260	5.99	6.55	6.60
总长度	D	.309	.315	.321	7.85	8.00	8.15
外露金属衬底长度	D2	.236	.258	.260	5.99	6.55	6.60
触点宽度	B	.008	.013	.013	0.20	0.33	0.35
触点长度	L	.014	.016	.019	0.35	0.40	0.48

*控制参数

注:

1. BSC: 基本尺寸。理论上显示的是没有公差精确值。

参见ASME Y14.5M

2. REF: 参考尺寸, 通常无公差, 仅供参考。

参见ASME Y14.5M

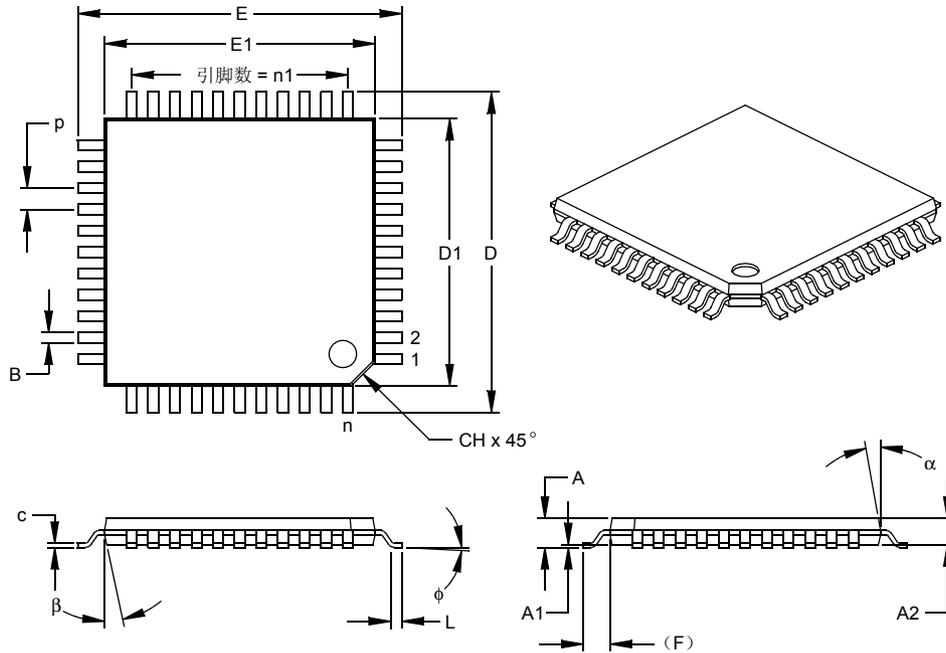
外露金属衬底尺寸随管芯叶片大小而变化。

等同于JEDEC号: MO-220

图号: C04-103, 修订于10-14-05

PIC18F45J10 系列

44 引脚塑封薄型正方扁平封装 (PT) —— 主体 10x10x1 mm, 引脚形式 1.0/0.10 mm (TQFP)



尺寸范围	单位	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n		44			44	
引脚间距	p		.031			0.80	
每侧引脚数	n1		11			11	
总高度	A	.039	.043	.047	1.00	1.10	1.20
塑模封装厚度	A2	.037	.039	.041	0.95	1.00	1.05
悬空间隙 §	A1	.002	.004	.006	0.05	0.10	0.15
底脚长度	L	.018	.024	.030	0.45	0.60	0.75
底脚投影长度 (参考)	(F)		.039		1.00		
底脚倾角	φ	0	3.5	7	0	3.5	7
总宽度	E	.463	.472	.482	11.75	12.00	12.25
总长度	D	.463	.472	.482	11.75	12.00	12.25
塑模封装宽度	E1	.390	.394	.398	9.90	10.00	10.10
塑模封装长度	D1	.390	.394	.398	9.90	10.00	10.10
引脚厚度	c	.004	.006	.008	0.09	0.15	0.20
引脚宽度	B	.012	.015	.017	0.30	0.38	0.44
引脚 1 处角斜面	CH	.025	.035	.045	0.64	0.89	1.14
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数
§ 重要特性

注：
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254 毫米)。
等同于 JEDEC 号: MS-026
图号: C04-076

PIC18F45J10 系列

注:

附录 A: 版本历史

版本 A (2005 年 3 月)

PIC18F45J10 系列器件数据手册的最初版本。

附录 B: 高档器件系列间的移植

PIC18F45J10 系列和 PIC18F4520 系列器件的功能和功能力件组非常相似。然而，当在不同器件系列间移植应用程序以达到新的设计目的时应注意器件系列间一些潜在的重要差异。表 B-1 中汇总了这些差异。本章随后将详细讨论对移植有重大影响的差异。

表 B-1: PIC18F4520 和 PIC18FXXXX 系列间明显的差异

特性	PIC18FXXXX 系列	PIC18F4520 系列
工作频率	40 MHz @ 2.15V	40 MHz @ 4.2V
供电电压	2.0V – 3.6V	2.0V – 5.5V
工作电流	低	更低
程序存储器耐久性	1,000 次擦写周期 (典型值)	100,000 次擦写周期 (典型值)
25 mA 的 I/O 灌 / 拉电流	仅 PORTB 和 PORTC	所有端口
I/O 引脚上的输入电压容差	仅数字引脚上为 5.5V	所有 I/O 引脚上均为 VDD
I/O	32	36
上拉	PORTB	PORTB
振荡器选项	有限选项 (EC、HS 和固定为 32 kHz 的 INTRC)	更多选项 (EC、HS、XT、LP、RC、PLL 和可变频率的 INTRC)
程序存储器数据保存期	10 年 (最少)	40 年 (最少)
编程时间 (标称值)	156 μs/B (10 ms/64 字节块)	15.6 μs/B (1 ms/64 字节块)
编程入口	低电压, 密钥序列	VPP 和 LVP
代码保护	单个存储区、全部或没有	多个代码保护存储区
配置字	存储在程序存储器空间的最后 4 个字中	存储在 300000h 开始的配置空间中
从休眠状态唤醒的起振时间	200 μs (典型值)	10 μs (典型值)
上电延时定时器	始终开启	可配置
数据 EEPROM	没有	有
BOR	简单 BOR ⁽¹⁾	可编程 BOR
LVD	没有	有
A/D 校准	必需	无需
在线仿真	不可用	可用
TMR3	不可用	可用
第二个 MSSP	有 ⁽²⁾	没有

注 1: PIC18LFXXJ10 器件不可使用欠压复位。

注 2: 仅 40/44 引脚器件上存在。

PIC18F45J10 系列

B.1 功耗要求差异

PIC18F45J10 系列和 PIC18F4520 器件系列最显著的差异在于功耗要求。PIC18F45J10 系列器件设计为用于规模较小的应用；因此它的最大电压较低，泄漏电流较大。

PIC18F45J10 系列器件的工作电压范围为 2.0V 至 3.6V。一个 VDD 引脚专用于内核逻辑供电引脚 (VDDCORE)。如第 23.0 节“电气规范”描述的那样，该引脚具有特定的电压和电容要求。

PIC18F45J10 系列器件的电流规范尚未确定。

B.2 引脚差异

PIC18F45J10 系列和 PIC18F4520 系列在引脚上存在着一些差异：

- 输入电压容差
- 输出电流的能力
- 可用 I/O 数目

PIC18F45J10 系列上仅具有数字输入功能的引脚的电压容差最高可达 5.5V，因而可以接受高于 VDD 的电压。第 9.0 节“I/O 端口”中的表 9-1 列出了完整的信息。

除了输入差异外还存在着输出差异。并非所有的 I/O 引脚都具有相同的拉 / 灌电流能力。只有 PORTB 和 PORTC 支持 25 mA 的拉 / 灌电流，而 PIC18F4520 的所有输出引脚都具有该输出能力。第 9.0 节“I/O 端口”中的表 9-2 列出了每个端口输出能力的信息。

PIC18F45J10 系列器件引脚功能的实现方式也有所不同。首先，OSC1/OSC2 振荡器引脚严格专用于外部振荡器功能，不可以像 PIC18F4520 器件那样将这些引脚重新分配给 I/O (RA6 或 RA7)。其次，MCLR 引脚只专用于 MCLR，不能被配置为输入引脚 (RE3)。最后一点是，PIC18F45J10 系列器件中不存在 RA4 引脚。

在 PIC18F4520 和 PIC18F45J10 系列器件之间移植时应注意所有的这些引脚差异（包括电源引脚的差异）。

B.3 振荡器差异

PIC18F4520 系列器件比 PIC18F45J10 系列器件具有更广泛的振荡器选择范围。PIC18F45J10 振荡器模式主要限制在 HS 和 EC 模式下。

此外，PIC18F45J10 系列具有一个仅可输出固定的 32 kHz 频率信号的内部 RC 振荡器，不支持 PIC18F4520 系列器件所支持的更高频率的 RC 模式。

B.4 外设

PIC18F45J10 系列能够在低至 2.15V 的电压下以 40 MHz 运行，而 PIC18F4520 系列仅限于在 4.2 +V 的电压下以 40 MHz 运行。

当在 PIC18F45J10 系列和 PIC18F4520 系列间移植时还必须考虑外设差异。

- **数据 EEPROM:** PIC18F45J10 系列器件没有此模块。
- **BOR:** PIC18F45J10 系列器件的 BOR 不可编程。通过使用内部稳压器提供简单的欠压复位功能 (PIC18LFXXJ10 器件不具备此功能)。
- **LVD:** PIC18F45J10 系列器件没有此模块。
- PIC18F45J10 系列没有 Timer3 (TMR3) 模块。
- T0CKI/C1OUT 引脚已从 RA4 移至 RB5。
- PIC18F45J10 系列中的 40/44 引脚器件具有第二个 MSSP 模块 (RD0:RD3)。

索引

A

A/D	209
A/D 转换器中断, 配置	213
ADCAL 位	216
ADCON0 寄存器	209
ADCON1 寄存器	209
ADCON2 寄存器	209
ADRESH 寄存器	209, 212
ADRESL 寄存器	209
采集要求	214
计算所需的最小采集时间	214
校准	216
配置模拟端口引脚	216
配置模块	213
使用 CCP2 触发信号	218
特殊事件触发信号 (CCP)	218
特殊事件触发器 (ECCP)	132
相关的寄存器	218
选择和配置采集时间	215
在功耗管理模式下的操作	216
转换	217
转换器特性	328
转换时钟 (TAD)	215
转换状态 (GO/DONE 位)	212
AC (时序) 特性	311
参数符号	311
器件时序规范的负载条件	312
时序条件	312
温度和电压规范	312
ACKSTAT	176
ACKSTAT 状态标志	176
ADCAL 位	216
ADCON0 寄存器	209
GO/DONE 位	212
ADCON1 寄存器	209
ADCON2 寄存器	209
ADDFSR	284
ADDLW	247
ADDLWLNK	284
ADDWF	247
ADDWFC	248
ADRESH 寄存器	209
ADRESL 寄存器	209, 212
ANDLW	248
ANDWF	249

B

BC	249
BCF	250
BF	176
BF 状态标志	176
BN	250
BNC	251
BNN	251
BNOV	252
BNZ	252
BOR	参见欠压复位
BOV	255
BRA	253
BRG	参见波特率发生器
BSF	253
BTFSC	254
BTFSS	254

BTG	255
BZ	256
版本历史	343
保护配置寄存器	240
比较 (CCP 模块)	126
CCPRx 寄存器	126
Timer1 模式选择	126
软件中断	126
特殊事件触发信号	218
特殊事件触发器	126
相关的寄存器	127
引脚配置	126
比较 (ECCP 模块)	132
特殊事件触发器	132
比较器	219
参考	
内部信号	221
外部信号	221
参考电压	221
复位的影响	222
工作原理	221
模拟输入连接注意事项	223
配置	220
输出	221
休眠期间的工作	222
相关的寄存器	223
响应时间	221
在休眠模式下的操作	222
中断	222
比较器参考电压模块	225
复位的影响	226
连接注意事项	226
配置	225
精度和误差	226
相关的寄存器	227
休眠期间的操作	226
比较器规范	310
编程, 器件指令	241
变更通知客户服务	355
表指针操作 (表)	70
表读 / 表写	50
表读操作	67
表写操作	68
并行从动端口 (PSP)	103, 109
CS (片选)	109
PORTD	109
RD (读输入)	109
WR (写输入)	109
相关的寄存器	110
选择 (PSPMODE 位)	103
波特率发生器	172
捕捉 / 比较 / PWM (CCP)	123
CCP 模块和定时器资源	124
CCPRxH 寄存器	124
CCPRxL 寄存器	124
ECCP1/CCP1 和 CCP2 模式——定时器资源	124
比较模式	参见比较
捕捉模式	参见捕捉
模块配置	124
捕捉 (CCP 模块)	125
CCP 引脚配置	125
CCPRxH:CCPRxL 寄存器	125

PIC18F45J10 系列

软件中断	125	使用 Timer1 中断服务实现实时时钟	119
相关的寄存器	127	写入闪存程序存储器	74
预分频器	125	在捕捉预分频器之间进行切换	125
捕捉 (ECCP 模块)	132	装载 SSP1BUF (SSP1SR) 寄存器	148
C		电气规范	297
C 编译器		读者反馈表	356
MPLAB C18	292	对标准 PIC 指令的影响	288
MPLAB C30	292	堆栈满 / 下溢复位	50
CALL	256	E	
CALLW	285	EUSART	
CLRF	257	波特率发生器	
CLRWDT	257	在功耗管理模式下的操作	191
COMF	258	波特率发生器 (BRG)	191
CPFSEQ	258	波特率, 异步模式	193
CPFSGT	259	波特率误差, 计算	192
CPFSLT	259	采样	191
CPU 的特殊功能	229	高波特率选择位 (BRGH 位)	191
参考电压规范	310	相关的寄存器	192
程序校验和代码保护	240	自动波特率检测	195
程序存储器		异步模式	197
查找表	50	12 位间隔字符发送和接收	202
复位矢量	47	发送器	197
和扩展的指令集	65	接收器	199
映射和堆栈 (框图)	47	设置带有地址检测功能的 9 位模式	199
中断矢量	47	同步间隔字符自动唤醒	200
指令	52	相关寄存器, 发送	198
双字	52	相关寄存器, 接收	200
程序计数器	48	同步从动模式	206
PCL、PCH 和 PCU 寄存器	48	发送	206
PCLATH 和 PCLATU 寄存器	48	接收	207
从动模式选择 (SSx)	145	相关寄存器, 发送	206
串行时钟, SCKx	145	相关寄存器, 接收	207
串行数据输出 (SDOx)	145	同步主控模式	203
串行数据输入 (SDIx)	145	发送	203
串行外设接口	参见 SPI 模式	接收	205
存储器编程要求	309	相关寄存器, 发送	204
存储器构成	47	相关寄存器, 接收	205
程序存储器	47	F	
数据存储器	53	FSCM	参见故障保护时钟监视器
存储区选择寄存器 (BSR)	53	返回地址堆栈	48
D		返回堆栈指针 (STKPTR)	49
DAW	260	封装信息	333
DC 和 AC 特性图表	331	标识	333
DCFSNZ	261	复位	37, 229
DECF	260	RESET 指令	37
DECFSZ	261	堆栈满复位	37
代码保护	229	堆栈下溢复位	37
代码示例		功耗管理模式下的 MCLR 复位	37
16 × 16 有符号乘法程序	78	看门狗定时器 (WDT) 复位	37
16 × 16 无符号乘法程序	78	可编程欠压复位 (BOR)	37
8 × 8 有符号乘法程序	77	欠压复位 (BOR)	229
8 × 8 无符号乘法程序	77	上电复位 (POR)	37, 229
擦除闪存程序存储器行	72	上电延时定时器 (PWRT)	229
初始化 PORTA	94	正常工作状态下的 MCLR 复位	37
初始化 PORTB	97	振荡器起振定时器 (OST)	229
初始化 PORTC	100	G	
初始化 PORTD	103	GOTO	262
初始化 PORTE	106	功耗管理模式	31
读取一个闪存程序存储器字	71	和多条 Sleep 命令	32
将 STATUS、WREG 和 BSR 寄存器的		和 A/D 运行	216
值保存在 RAM 中	91	和 EUSART 的工作原理	191
快速寄存器堆栈	50	和 PWM 工作原理	143
使用偏移量计算 GOTO	50	和 SPI 模式的工作原理	153
使用间接寻址清零 RAM (存储区 1) 的方法	61		

汇总 (表)	31	启动条件时序	174
进入	31	重复启动条件时序	175
空闲模式	34	总线冲突	
PRI_IDLE	35	在重复启动条件期间	183
RC_IDLE	36	停止条件期间	184
SEC_IDLE	35	INCF	262
退出空闲和休眠模式		INCFSZ	263
通过 WDT 超时	36	INFSNZ	263
不需要振荡器起振延迟	36	INTCON 寄存器	81
通过复位	36	INTOSC, INTRC	参见内部振荡器电路
时钟转换和状态指示位	32	IORLW	264
休眠模式	34	IORWF	264
选择	31	IPR 寄存器	88
运行模式	32	J	
PRI_RUN	32	寄存器	
RC_RUN	33	ADCON0 (A/D 控制 0)	209
SEC_RUN	32	ADCON1 (A/D 控制寄存器 1)	210
功耗管理模式对各种时钟源的影响	28	ADCON2 (A/D 控制寄存器 2)	211
公式		BAUDCON (波特率控制)	190
A/D 采集时间	214	CCP1CON (增强型捕捉 / 比较	
A/D 最小充电时间	214	/PWM 控制寄存器 1)	131
故障保护时钟监视器	229, 238	CCPxCON (标准捕捉 / 比较 / PWM 控制)	123
POR 或从休眠中唤醒	239	CMCON (比较器控制)	219
功耗管理模式下的中断	239	CONFIG1H (配置寄存器 1 的高字节)	231
振荡器故障期间的 WDT	238	CONFIG1L (配置寄存器 1 的低字节)	231
固件指令	241	CONFIG2H (配置寄存器 2 的高字节)	232
H		CONFIG2L (配置寄存器 2 的低字节)	232
汇编器		CONFIG3H (配置寄存器 3 的高字节)	233
MPASM 汇编器	292	CONFIG3L (配置寄存器 3 的低字节)	233
I		CVRCON (比较器参考电压控制)	225
I/O 端口	93	ECCP1AS (ECCP 自动关闭控制)	141
I/O 引脚排列说明		ECCP1DEL (PWM 死区延迟)	140
PIC18F24J10/25J10	12, 16	EECON1 (数据 EEPROM 控制 1)	69
I ² C 模式 (MSSP)		INTCON (中断控制)	81
I ² C 时钟速率和 BRG	172	INTCON2 (中断控制 2)	82
波特率发生器	172	INTCON3 (中断控制 3)	83
从动模式	159	IPR1 (外设中断优先级 1)	88
寻址	159	IPR2 (外设中断优先级 2)	89
发送	160	IPR3 (外设中断优先级 3)	89
接收	160	OSCCON (振荡器控制)	28
串行时钟 (SCKx/SCLx)	160	OSCTUNE (PLL 控制)	25
读 / 写信息位 (R/W 位)	159, 160	PIE1 (外设中断使能 1)	86
多主控机模式	180	PIE2 (外设中断使能 2)	87
多主控机通信、总线冲突与总线仲裁	180	PIE3 (外设中断使能 3)	87
复位的影响	180	PIR1 (外设中断请求 (标志) 1)	84
工作模式	159	PIR2 (外设中断请求 (标志) 2)	85
广播呼叫地址支持	169	PIR3 (外设中断请求 (标志) 3)	85
寄存器	155	RCON (复位控制)	38, 90
时钟延长	165	RCSTA (接收状态和控制寄存器)	189
10 位从动发送模式	165	SSPxCON1 (MSSPx 控制寄存器 1, I ² C 模式)	157
10 位从动接收模式 (SEN = 1)	165	SSPxCON1 (MSSPx 控制寄存器 1, SPI 模式)	147
7 位从动发送模式	165	SSPxCON2 (MSSPx 控制寄存器 2, I ² C 模式)	158
7 位从动接收模式 (SEN = 1)	165	SSPxSTAT (MSSPx 状态寄存器, I ² C 模式)	156
时钟仲裁	173	SSPxSTAT (MSSPx 状态寄存器, SPI 模式)	146
时钟同步与 CKP 位	166	STATUS	60
停止条件时序	179	STKPTR (堆栈指针)	49
相关的寄存器	185	T0CON (Timer0 控制寄存器)	111
休眠工作方式	180	T1CON (Timer1 控制寄存器)	115
应答序列时序	179	T2CON (Timer2 控制寄存器)	121
主控模式	170	TRISE (PORTE/PSP 控制)	107
波特率发生器	172	TXSTA (发送状态和控制寄存器)	188
发送	176	WDTCON (看门狗定时器控制)	235
工作模式	171	器件 ID 寄存器 1	234
接收	176	器件 ID 寄存器 2	234
		寄存器汇总	57-59

PIC18F45J10 系列

寄存器文件	55	SPI 主 / 从连接	149
计算 GOTO	50	SSPxBUF 寄存器	150
间接寻址	62	SSPxSR 寄存器	150
间隔字符 (12 位) 发送和接收	202	模块概述	145
校准 (A/D 转换器)	216	控制寄存器 (通用)	145
晶振 / 陶瓷谐振器	23	MULLW	268
绝对最大值	297	MULWF	268
K		脉宽调制	参见 PWM (CCP 模块) 和 PWM (ECCP 模块)
开发支持	291	模数转换器	参见 A/D
看门狗定时器 (WDT)	229, 235	默认系统时钟	27
编程注意事项	235	N	
振荡器故障期间	238	NEGF	269
相关的寄存器	235	NOP	269
控制寄存器	235	内部 RC 振荡器	
勘误表	6	与 WDT 一起使用	235
客户支持	355	内部互联模块	参见 I ² C 模式
客户通知服务	355	内部振荡器电路	26
快速操作存储区		P	
使用立即数变址寻址模式映射	65	PIC18F4520 和 PIC18F45J10 系列间明显的差异	343
快速寄存器堆栈	50	功耗要求	344
框图		外设	344
MSSP (I ² C 主控模式)	170	引脚	344
MSSP (SPI 模式)	145	振荡器选项	344
PIC18F24J10/25J10	10, 11	PICSTART Plus 开发编程器	294
波特率发生器	172	PIE 寄存器	86
读取闪存程序存储器	71	PIR 寄存器	84
对闪存程序存储器执行表写操作	73	PLL 倍频器	25
故障保护时钟监视器	238	ECPLL 振荡器模式	25
扩展的指令集		HSPLL 振荡器模式	25
ADDFSR	284	POP	270
ADDULNK	284	POR	参见上电复位
CALLW	285	PORTA	
MOVSF	285	LATA 寄存器	94
MOVSS	286	PORTA 寄存器	94
PUSHL	286	TRISA 寄存器	94
SUBFSR	287	相关的寄存器	96
SUBULNK	287	PORTB	
和使用 MPLAB 工具	290	LATB 寄存器	97
使用注意事项	288	PORTB 寄存器	97
语法	283	RB7:RB4 电平变化中断标志 (RBIF 位)	97
L		TRISB 寄存器	97
LFSR	265	相关的寄存器	99
立即数变址寻址		PORTC	
和标准的 PIC18 指令	288	LATC 寄存器	100
立即数寻址模式	288	PORTC 寄存器	100
M		RC3/SCK1/SCL1 引脚	160
Microchip 因特网网站	355	TRISC 寄存器	100
MOVF	265	相关的寄存器	102
MOVFF	266	PORTD	
MOVLB	266	LATD 寄存器	103
MOVLW	267	PORTD 寄存器	103
MOVSF	285	TRISD 寄存器	103
MOVSS	286	并行从动端口 (PSP) 功能	103
MOVWF	267	相关的寄存器	105
MPLAB ASM30 汇编器、链接器和库管理器	292	PORTE	
MPLAB ICD 2 在线调试器	293	LATE 寄存器	106
MPLAB ICE 2000 高性能通用在线仿真器	293	PORTE 寄存器	106
MPLAB ICE 4000 高性能通用在线仿真器	293	PSP 模式选择 (PSPMODE 位)	103
MPLAB PM3 器件编程器	293	TRISE 寄存器	106
MPLAB 集成开发环境软件	291	相关的寄存器	108
MPLINK 目标链接器 / MPLIB 目标库管理器	292	PRI_IDLE 模式	35
MSSP		PRI_RUN 模式	32
ACK 脉冲	159, 160	PSP	参见并行从动端口
		PUSH	270

PUSH 和 POP 指令	49	SPI 模式 (MSSP)	
PUSHL	286	SPI 时钟	150
PWM (CCP 模块)		从动选择同步	151
CCPR1H:CCPR1L 寄存器	133	从动模式	151
操作的设置	129	从动模式选择	145
当 TMR2 与 PR2 匹配时	128, 133	串行时钟	145
频率 / 分辨率示例	129, 134	串行数据输出	145
相关的寄存器	130	串行数据输入	145
占空比	128, 134	典型连接	149
周期	128, 133	复位的影响	153
自动关闭 (仅 CCP1)	129	工作方式	148
PWM (ECCP 模块)	133	时钟速度和模块相互关系	153
PWM 操作的设置	143	使能 SPI I/O	149
半桥输出模式应用示例	136	相关的寄存器	154
半桥模式	136	在功耗管理模式下的工作方式	153
复位的影响	143	主 / 从连接	149
可编程死区延迟	140	主控模式	150
启动注意事项	142	总线模式兼容性	153
全桥输出模式下的方向更改	138	SSPOV	176
全桥应用示例	138	SSPOV 状态标志	176
全桥模式	137	SSPxSTAT 寄存器	
输出关系 (低电平有效)	135	___ R/W 位	159, 160
输出关系 (高电平有效)	135	SSx	145
输出配置	134	SUBFSR	287
在功耗管理模式下的操作	143	SUBFWB	276
在使用故障保护时钟监视器时的工作原理	143	SUBLW	277
增强型 PWM 自动关闭	140	SUBULNK	287
配置位	229	SUBWF	277
Q		SUBWFB	278
Q 时钟	129, 134	SWAPF	278
器件概述	7	双字指令	
其他特殊功能	8	示例情形	52
新的内核功能	7	双速启动	229, 237
特性 (表)	9	所有寄存器的初始化状态	43-46
系列中各产品的具体信息	8	闪存程序存储器	67
欠压复位 (BOR)	39	表指针边界	70
和片内稳压器	236	擦除	72
在休眠模式下禁止	39	擦除序列	72
R		代码保护时的操作	75
RAM	参见数据存储器	读	71
RBIF 位	97	表读和表写	67
RC_IDLE 模式	36	基于操作的表指针边界	70
RC_RUN 模式	33	控制寄存器	68
RCALL	271	EECON1 和 EECON2	68
RCON 寄存器		TABLAT (表锁存器)	70
初始化期间的位状态	42	TBLPTR (表指针)	70
RESET	271	相关的寄存器	75
RETFIE	272	写操作序列	73
RETLW	272	写入	73
RETURN	273	避免误写	75
RLCF	273	写校验	75
RLNCF	274	意外终止	75
RRCF	274	闪存配置字	229
RRNCF	275	上电复位 (POR)	39
软件模拟器 (MPLAB SIM)	292	上电延时定时器 (PWRT)	40
S		延时时序	40
SCKx	145	上电延时	29
SDIx	145	上电延迟定时器 (PWRT)	29, 40
SDOx	145	时钟源	26
SEC_IDLE 模式	35	复位时的默认系统时钟	27
SEC_RUN 模式	32	使用 OSCCON 寄存器进行选择	27
SETF	275	时序图	
SLEEP	276	A/D 转换	328
		BRG 溢出时序	196
		CLKO 和 I/O	315

PIC18F45J10 系列

EUSART 同步发送 (主控 / 从动)	327	异步发送 (背对背)	198
EUSART 同步接收 (主控 / 从动)	327	异步接收	200
I ² C 从动模式 (7 位接收, SEN = 0)	161	应答序列	179
I ² C 从动模式广播呼叫地址时序 (7 位或 10 位地址模式)	169	在接近 100% 占空比时 PWM 更改方向	139
I ² C 主控模式 (7 位接收)	178	正常工作模式下的自动唤醒位 (WUE)	201
I ² C 主控模式 (7 位或 10 位地址发送)	177	主控 SSP I ² C 总线启动 / 停止位	325
I ² C 总线启动 / 停止位	323	主控 SSP I ² C 总线数据	325
I ² C 总线数据	323	自动波特率计算	196
I ² C 停止条件接收或发送模式	179	时序图和规范	
PWM 方向更改	139	A/D 转换要求	329
PWM 输出	128	AC 特性	
PWM 自动关闭 (PRSEN = 0, 禁止自动重启)	142	内部 RC 精度	314
PWM 自动关闭 (PRSEN = 1, 使能自动重启)	142	CLKO 和 I/O 要求	315
SPI 从动模式示例 (CKE = 0)	321	EUSART 同步发送要求	327
SPI 从动模式示例 (CKE = 1)	322	EUSART 同步接收要求	327
SPI 主控模式示例 (CKE = 0)	319	I ² C 总线启动 / 停止位要求 (从动模式)	323
SPI 主控模式示例 (CKE = 1)	320	I ² C 总线数据要求 (从动模式)	324
SPI 模式 (从动模式, CKE = 0)	152	外部时钟要求	313
SPI 模式 (从动模式, CKE = 1)	152	PLL 时钟	314
SPI 模式 (主控模式)	150	SPI 从动模式要求示例 (CKE = 1)	322
Timer0 和 Timer1 外部时钟	317	SPI 模式要求示例 (从动模式, CKE = 0)	321
半桥 PWM 输出	136	SPI 模式要求示例 (主控模式, CKE = 0)	319
并行从动端口 (PSP) 写	110	SPI 模式要求示例 (主控模式, CKE = 1)	320
并行从动端口 (PSP) 读	110	Timer0 和 Timer1 外部时钟要求	317
捕捉 / 比较 / PWM (包括 ECCP 模块)	318	并行从动端口要求	318
重复启动条件	175	捕捉 / 比较 / PWM 要求 (包括 ECCP 模块)	318
重复启动条件期间的总线冲突 (情形 1)	183	复位、看门狗定时器、振荡器起振定时器、上电延时定 时器和欠压复位要求	316
重复启动条件期间的总线冲突 (情形 2)	183	主控 SSP I ² C 总线启动 / 停止位要求	325
从动同步	151	主控 SSP I ² C 总线数据要求	326
从休眠模式唤醒的转换	34	数据寻址模式	61
从空闲模式唤醒进入运行模式的转换	35	固有和立即数	61
从 RC_RUN 模式到 PRI_RUN 模式的转换	33	立即数变址寻址	
带有时钟仲裁的波特率发生器	173	影响的指令	63
到 RC_RUN 模式的转换	33	与使能了扩展的指令集的寻址模式对比	64
第一个启动位时序	174	间接寻址	61
发送和应答时的总线冲突	180	直接寻址	61
发送间隔字符时序	202	数据存储区	53
复位、看门狗定时器 (WDT)、振荡器起振定时器 (OST) 和上电延时定时器 (PWRT)	316	PIC18F24J10/44J10 映射	54
故障保护时钟监视器	239	存储区选择寄存器 (BSR)	53
缓慢上升时间 (MCLR 连接到 VDD, VDD 上升时间 > TPWRT)	41	和扩展的指令集	63
进入休眠模式转换	34	特殊功能寄存器	56
进入空闲模式的转换	35	通用寄存器	55
进入 SEC_RUN 模式的转换	32	快速操作存储区	55
启动条件期间的总线冲突 (仅 SDAx)	181	T	
启动条件期间的总线冲突 (SCLx = 0)	182	TBLRD	279
启动条件期间由 SDAx 仲裁引起的 BRG 复位	182	TBLWT	280
欠压复位 (BOR)	316	Timer0	111
全桥 PWM 输出	137	16 位读写模式	112
上电时的延时时序 (MCLR 连接到 VDD, VDD 上升时间 为 Tpwrt)	40	工作原理	112
上电时的延时时序 (MCLR 未连接到 VDD) 情形 2 ...	41	切换预分频器的分配	113
时钟 / 指令周期	51	时钟源选择 (T0CS 位)	112
时钟同步	166	时钟源边沿选择 (TOSE 位)	112
双速启动 (INTRC) 的时序过渡	237	相关的寄存器	113
停止条件期间的总线冲突 (情形 1)	184	溢出中断	113
停止条件期间的总线冲突 (情形 2)	184	预分频器	113
同步发送	203	预分频器	参预分频器, Timer0
同步发送 (由 TXEN 位控制)	204	预分频器选择 (T0PS2:T0PS0 位)	113
同步接收 (主控模式, 由 SREN 位控制)	205	预分频器分配 (PSA 位)	113
外部时钟 (除 PLL 之外的所有模式)	313	Timer1	115
休眠模式下的自动唤醒位 (WUE)	201	16 位读 / 写模式	117
异步发送	198	TMR1H 寄存器	115
		TMR1L 寄存器	115
		使用 ECCP/CCP 特殊事件触发信号来复位 Timer1 ..	118
		工作原理	116

特殊事件触发器 (ECCP)	132	RC5/SDO1	15, 19
相关的寄存器	119	RC6/TX/CK	15, 19
溢出中断	115	RC7/RX/DT	15, 19
用作时钟源	117	RD0/PSP0/SCK2/SCL2	20
振荡器	115, 117	RD1/PSP1/SDI2/SDA2	20
布局注意事项	118	RD2/PSP2/SDO2	20
振荡器, 作为辅助时钟	26	RD3/PSP3/SS2	20
中断	118	RD4/PSP4	20
作为实时时钟使用	118	RD5/PSP5/P1B	20
Timer2	121	RD6/PSP6/P1C	20
PR2 寄存器	128, 133	RD7/PSP7/P1D	20
当 TMR2 与 PR2 匹配时中断	128, 133	RE0/RD/AN5	21
工作原理	121	RE1/WR/AN6	21
输出	122	RE2/CS/AN7	21
中断	122	VDD	15, 21
相关的寄存器	122	VDDCORE/VCAP	15, 21
TRISE 寄存器		Vss	15, 21
PSPMODE 位	103	硬件乘法器	77
TSTFSZ	281	工作原理	77
TXSTA 寄存器		性能比较	77
BRGH 位	191	简介	77
特殊功能寄存器	56	预分频器	
映射	56	Timer2	134
特殊事件触发器	参见比较 (ECCP/CCP 模块)	预分频器, Timer0	113
特殊事件触发器	参见比较 (ECCP 模块)	预分频器, Timer2	129
同步间隔字符自动唤醒	200	框图	
W		16 位模式 Timer0	112
WCOL	174, 175, 176, 179	8 位模式 Timer0	112
WCOL 状态标志	174, 175, 176, 179	A/D	212
WWW, 在线支持	6	EUSART 发送	197
WWW 地址	355	EUSART 接收	199
外部时钟输入 (EC 模式)	24	MSSP (I ² C 模式)	155
稳压器 (片内)	236	PLL	25
X		PORTD 和 PORTE (并行从动端口)	109
XORLW	281	PWM 操作 (简化)	128
XORWF	282	Timer1	116
休眠模式		Timer1 (16 位读 / 写模式)	116
OSC1 和 OSC2 引脚状态	29	Timer2	122
Y		比较器参考电压输出缓冲示例	227
引脚功能		比较器参考电压模块	226
MCLR	12, 16	比较器输出	222
OSC1/CLKI	12, 16	比较器模拟输入典型电路	223
OSC2/CLKO	12, 16	比较器 I/O 工作模式	220
RA0/AN0	13, 17	比较模式工作	126
RA1/AN1	13, 17	捕捉模式工作	125
RA2/AN2/VREF-/CVREF	13, 17	单个比较器	221
RA3/AN3/VREF+	13, 17	模拟输入典型电路	213
RA5/AN4/SS1/C2OUT	13, 17	片上复位电路	37
RB0/INT0/FLT0/AN12	14, 18	器件时钟	26
RB1/INT1/AN10	14, 18	通用 I/O 端口的操作	93
RB2/INT2/AN8	14, 18	外部上电复位电路 (VDD 慢速上电)	39
RB3/AN9/CCP2	14, 18	增强型 PWM	133
RB4/KBI0/AN11	14, 18	中断逻辑	80
RB5/KBI1/C1OUT	18	Z	
RB5/KBI1/T0CKI/C1OUT	14	在线串行编程 (ICSP)	229, 240
RB6/KBI2/PGC	14, 18	在线调试器	240
RB7/KBI3/PGD	14, 18	增强型捕捉 / 比较 / PWM (ECCP)	131
RC0/T1OSO/T1CKI	15, 19	ECCP1 模式的引脚配置	132
RC1/T1OSI/CCP2	15, 19	PWM 模式	参见 PWM (ECCP 模块)
RC2/CCP1	15	标准 PWM 模式	132
RC2/CCP1/P1A	19	捕捉和比较模式	132
RC3/SCK1/SCL1	15, 19	捕捉模式	参见捕捉 (ECCP 模块)
RC4/SDI1/SDA1	15, 19	定时器资源	132
		输出和配置	132
		相关的寄存器	144

PIC18F45J10 系列

增强型通用同步 / 异步收发器 (EUSART) 参见 EUSART	DCFSNZ	261	
增强型 PWM 模式.....	133	DECFSZ	260
栈顶访问	48	DECFSZ	261
振荡器选择	229	GOTO	262
振荡器, Timer1	115	INCF	262
振荡器配置	23	INCFSZ	263
EC	23	INFSNZ	263
ECPLL	23	IORLW	264
HS	23	IORWF	264
HS 模式	23	LFSR	265
HSPLL	23	MOVF	265
INTRC	23	MOVFF	266
内部振荡器电路	26	MOVLB	266
振荡器起振定时器 (OST)	29	MOVLW	267
振荡器切换	26	MOVWF	267
振荡器转换	27	MULLW	268
直接寻址	62	MULWF	268
中断	79	NEGF	269
中断, 标志位		NOP	269
电平变化中断 (RB7:RB4) 标志 (RBIF 位)	97	通用格式	243
中断的现场保护	91	POP	270
中断源	229	PUSH	270
A/D 转换完成	213	RCALL	271
INTn 引脚	91	RESET	271
PORTB 电平变化中断	91	RETFIE	272
TMR0	91	RETLW	272
TMR0 溢出	113	RETURN	273
TMR1 溢出	115	RLCF	273
比较完成 (CCP)	126	RLNCF	274
捕捉完成 (CCP)	125	RRCF	274
当 TMR2 与 PR2 匹配时中断 (PWM)	128, 133	RRNCF	275
电平变化中断 (RB7:RB4)	97	SETF	275
指令流 / 流水线	51	SETF (立即数变址寻址模式)	289
指令周期	51	SLEEP	276
时钟分配	51	SUBFWB	276
指令集	241	SUBLW	277
ADDLW	247	SUBWF	277
ADDWF	247	SUBWFB	278
ADDWF (立即数变址寻址模式)	289	SWAPF	278
ADDWFC	248	TBLRD	279
ANDLW	248	TBLWT	280
ANDWF	249	TSTFSZ	281
BC	249	XORLW	281
BCF	250	XORWF	282
BN	250	标准指令	241
BNC	251	操作码字段说明	242
BNN	251	扩展的指令集	283
BNOV	252	主复位 (MCLR)	39
BNZ	252	主控同步串行端口 (MSSP)	参见 MSSP
BOV	255		
BRA	253		
BSF	253		
BSF (立即数变址寻址模式)	289		
BTFSC	254		
BTFSS	254		
BTG	255		
BZ	256		
CALL	256		
CLRF	257		
CLRWDT	257		
COMF	258		
CPFSEQ	258		
CPFSGT	259		
CPFSLT	259		
DAW	260		

注:

PIC18F45J10 系列

注:

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息:

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请登录 Microchip 网站 www.microchip.com, 点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持
- 开发系统信息热线

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://support.microchip.com> 获得网上技术支持。

PIC18F45J10 系列

读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。

请填写以下信息，并从下面各方面提出您对本文档的意见。

致： TRC 经理 总页数 _____
关于： 读者反馈
发自： 姓名 _____
公司 _____
地址 _____
国家 / 省份 / 城市 / 邮编 _____
电话 (_____) _____ 传真 (_____) _____

应用 (选填):

您希望收到回复吗? 是____ 否____

器件: PIC18F45J10 系列 文献编号: DS39682A_CN

问题

1. 本文档中哪些部分最有特色?

2. 本文档是否满足了您的软硬件开发要求? 如何满足的?

3. 您认为本文档的组织结构便于理解吗? 如果不便于理解, 那么问题何在?

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

5. 您认为本文档中可以删减哪些内容, 而又不会影响整体使用效果?

6. 本文档中是否存在错误或误导信息? 如果存在, 请指出是什么信息及其具体页数。

7. 您认为本文档还有哪些方面有待改进?

PIC18F45J10 系列产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或销售办事处联系。

器件编号	X	/XX	XXX
器件	温度范围	封装	模式
器件	PIC18F24J10/25J10 ⁽¹⁾ , PIC18F44J10/45J10 ⁽¹⁾ , PIC18F24J10/25J10T ⁽²⁾ 和 PIC18F44J10/45J10T ⁽²⁾ ; V _{DD} 范围为 4.2V 至 5.5V PIC18LF24J10/25J10 ⁽¹⁾ , PIC18LF44J10/45J10 ⁽¹⁾ , PIC18LF24J10/25J10T ⁽²⁾ 和 PIC18LF44J10/45J10T ⁽²⁾ ; V _{DD} 范围为 2.0V 至 5.5V		
温度范围	I 表示 -40°C 至 +85°C (工业级) E 表示 -40°C 至 +125°C (扩展级)		
封装	PT 表示 TQFP (薄型正方扁平封装) SO 表示 SOIC SP 表示窄型塑封 DIP P 表示 PDIP ML 表示 QFN		
模式	QTP、SQTP、编码或特殊要求 (空白为其他情况)		

示例:

- a) PIC18LF45J10-I/P 301 表示工业级温度、PDIP 封装、扩展级 V_{DD} 范围和 QTP 模式 # 301。
- b) PIC18LF24J10-I/SO 表示工业级温度、SOIC 封装和扩展级 V_{DD} 范围。
- c) PIC18LF44J10-I/P 表示工业级温度、PDIP 封装和普通 V_{DD} 范围。

注 1: F 表示标准电压范围
 LF 表示宽电压范围

注 2: T 表示卷带式封装 (仅 TQFP 封装)。



MICROCHIP

全球销售及服务中心

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://support.microchip.com>
网址: www.microchip.com

亚特兰大 Atlanta
Alpharetta, GA
Tel: 1-770-640-0034
Fax: 1-770-640-0307

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 Kokomo
Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣何塞 San Jose
Mountain View, CA
Tel: 1-650-215-1444
Fax: 1-650-961-0286

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

中国 - 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

中国 - 福州
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

印度 India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

韩国 Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

10/31/05